DETECTION OF PARAMETRIC HARDWARE TROJANS

By

Raeesa Naseem

A thesis submitted to the Faculty of Information Security Department, Military College of Signals, National University of Science and Technology, Pakistan in partial fulfillment of the requirements for the degree of Master of Science in Information Security

April 2016

ABSTRACT

The eminence of third-party-foundry business practice saves a lot of cost by utilizing the economy of scale. While this practise saves a lot of cost by utilizing the economy of scale, it exposes these chips to various threats that includes insertion of hardware trojan.

ICs form the core for the computing and communication systems used in contemporary personal, commercial and government affairs. The repercussions are critical, considering the penetration of technology in military and commercial sectors. For this reason, granting trust in presence of unreliable third-party fabrication has become a major challenge. Hardware trojans and, consequently, defence-in-depth has therefore drawn a great attention in research and development programs.

The scope of this thesis is also to form a baseline for hardware trojan detection. To study the unexpected behaviour of ICs because of trojans, a taxonomy of different types of hardware trojans based on their various attributes is proposed.

The research is focused on Parametric Hardware Trojans (PHTs), the stealthiest emergence of HTs. It is very sophisticated trojan and it bypasses most of the widely used detection techniques. Detection of PHTs is therefore considered as a challenge and this is taken as an objective of this research. The research not only comprehends the detection techniques suitable for PHTs but also analyses the effectiveness of Side Channel Analysis (SCA) to detect PHTs. SCA is used widely for attacking and compromising the security. In this research, this concept is conceived with a detective and defensive approach.

This research also contributes in the prevention and countermeasure techniques for PHTs. The proposed techniques can be used as a stand alone or as a combination, depending on the level of required security.

For this, a high profile target was studied that is Random Number Generator (RNG) of Intel's Ivy Bridge microprocessor. The concept of PHT is concieved from (1). To detect this trojan, effec-

tiveness of SCA is analysed. The trojan is visualized at Hardware Description Language (HDL) level and implemented on FPGA board. Different limitations were observed while performing SCA over FPGA development kits and two detection models were proposed in MATLAB and its effectiveness was analysed.

DEDICATION

I dedicate this thesis to my parents, family and friends who helped and supported me to achieve what I have today.

ACKNOWLEDGMENT

In the name of Allah, the Most Gracious and the Most Merciful

All praises to Allah for the strengths and His blessing in completing this thesis. I am extremely grateful and obliged to Dr. Mehreen Afzal for providing me her valuable guidance and professional advice during the course of this study. Her full devotion and dedication towards my research work, her constructive comments and suggestions throughout the experimental analysis were major source of encouragement for me in successfully completing this research work. I will also like to thank Dr Adnan Rashdi, my thesis co-supervisor, for extending his fullest cooperation and giving me time to time suggestion, resources and guidance for experimental results.

I am grateful to my guidance committee members; Dr Ashraf Masood, Dr Babar Aslam, and Mian Mohammad Waseem Iqbal for their support, evaluation, cooperation and giving me the required knowledge for this study. i am grateful to Mr Rashid Ali, Mr Roveed Ahmed, and Mrs Saima Gul, for extending me their fullest cooperation and support during the course of my implementation phase.

Finally, I am obliged to faculty and staff members of Image Processing Center (IPC) Lab, Military College of Signals for providing me favourable and conducive research environment.

Acknowledgements would not be complete without extending gratitude to my parents, siblings and friends. They have always stood by my dreams and aspirations and have been a great source of inspiration and encouragement.

Raeesa Naseem

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction

This chapter gives an overview of the aim and objective of the research that is carried out. It starts with a brief overview of hardware trojans that is discussed later in detail in next chapter. Second section emphasizes on need for research in this topic, followed by highlighting its importance and relevance for the national needs. The problem statement is followed by the objectives of thesis. The methodology adopted for achieving these goals is also mentioned next and the chapter concludes with thesis organization section.

1.2 Overview of Hardware Trojans

The ever-increasing cost of manufacturing Integrated Circuits (ICs) in small-scale CMOS technology has led to the eminence of third party foundry business practice (2). While this saves a lot of cost by utilizing the economy of scale, it also exposes the chips, by authentic designers, to threats such as hardware trojan insertion, unlicensed IP handling, and IP piracy. Since ICs form the core for the computing and communication systems which are used in contemporary personal, commercial, and government affairs, their exposure to such threats endangers the full systems built upon them. The repercussions are critical, considering the penetration of technology in military and commercial sectors. Hardware trojans can therefore also be used as cyber weapons just like other traditional weapons in cyber warfare (2). Therefore, granting trust in presence of

unreliable third-party fabrication has become a major challenge.

Hardware trojans and, consequently, defense-in-depth has therefore drawn a great attention in research and development (RD) programs. It is no more a myth and a lot of years of research have already been dedicated to it in developed countries.

Hardware Trojans are the intentional modification or manipulation of Application Specific Integrated Circuits ASICs, commercial-off-the-shelf (COTS) parts, microprocessors, microcontrollers, network processors, or digital-signal processors (DSPs) in such a way that the expected output of the device or circuit is undesirably altered or leaked.

An indepth study of hardware trojans, its various types and different inserting techniques can be studied in later chapters.

1.3 Need for Research

Hardware trojans in ICs is considered as a myth in most of the developing countries. Since ICs form the core for the computing and communication systems used in contemporary personal, commercial and government affairs, their exposure to such threats endangers the complete systems built on them. The repercussions are critical, considering the penetration of technology in military and commercial sectors. Hardware trojans can also be used as cyberweapons just like other traditional weapons (2). The possible adversaries are likely to be financially and technologically advanced and thus, intelligent attacks are possible. Therefore, granting trust in presence of unreliable third-party fabrication has become a major challenge.

Hardware trojans and, consequently, defense-in-depth has therefore drawn a great attention in research and development programs. It is no more a myth and a lot of years of research have already been dedicated to it in developed countries.

Another standing challenge for non-invasive IC testing and trojan detection is dealing with the

increasing complexity and scale of state-of-the-art technology. The conventional parametric IC testing methods have a limited effect for addressing trojan related problems. This is because of the hidden triggering mechanism of trojans which ,the logic-based testing methods, are unlikely to trigger to distinguish the malicious alterations. It is also hard to differentiate between the physical and functional properties which are either because of characteristic deviations because of the process variations or, the alterations due to the trojan insertion. Destructive tests and IC reverse-engineering are slow and expensive. Whereas classic testing methods are insufficient for such trojan's detection. What complicates the problem even more is that the space for possible changes by the adversary is large.

While studying about the above two considerations, very less work and knowledge is found in our local RD departments. It was therefore felt a need for research to form a baseline of hardware trojans and its detection. Much more research and development is needed for finding scalable and cost-effective trojan testing and identification methods, establishing detection bounds, and improving statistical detection, calibration, and sensitivity analysis.

1.4 Relevance to National Needs

Understanding hardware trojans is critical to evaluate and protect hardware devices against it. This is a critical issue to be addressed in present era of cyber world.

Hardware trojans is a not an old concept and its thorough understanding can open multiple dimensions of threats and criteria of evaluating our devices. Today, many national organizations in general and military organizations in particular are heavily dependent on information technology. Embedded devices are integral component of most of the IT devices. Presence of such stealthy hardware trojans can do much of the damage such as leaking out secret information such as secret keys, which as a result, nullifies the effect of any cryptographic protocol used for integrity

or/and confidentiality. For instance, in 2010 the chip broker VisionTech was charged with selling fake chips, many of which were destined for safety and security critical systems such as high-speed train breaks, hostile radar tracking in F-16 fighter jets, and ballistic missile control systems (3). Therefore awareness and knowledge of such trojans helps in evaluation of embedded system's performance and the organization's decision of relying on it. Also finding effective way of detecting such trojans can prevent them from exposure to various attacks.

1.5 Problem Statement

Understanding hardware trojans is very important when developing next generation defensive or critical mechanisms for the development and deployment of electronics. This is a new emerging way of attack and a very serious problem for nations like us who are generally buyers of technology. Efforts in the direction of detection of Hardware Trojans (HTs) is the need of hour, however detection of hardware trojans is a big challenge. Relevant literature gives some techniques for detection and side channel analysis (SCA), which was once used for attacking Cryptographic hardware, is one of the most important mechanisms. Parametric Hardware Trojans (PHT) are most stealthy trojans and very limited literature is available on their detection. It is therefore imperative to explore the use of SCA for detection of PHTs in ICs. Many SCA models for attack and detection can be found in literature, however there appropriate applicability on PHTs is an important area of research. Moreover, there is very limited work and expertise available in this field in our country, thus to explore this area of cyber security in our academia has vital importance.

1.6 Objectives

While stating the problem statement, the objective of this research is to:

1. Study Parametric Hardware Trojans which are the stealthiest emergence of hardware trojans. This includes studying different insertion techniques of PHTs, possible attacks and its different detection methodologies.

2. Study and analyse different detection techniques with main focus on Side Channel Analysis (SCA).

3. Study dopant level hardware trojan on Intel RNG as a case study and propose SCA model for its detection.

4. FPGA based implementation of PHT and its detection though SCA for proof of concept.

5. Analysing different preventions and countermeasure for the studied trojan.

1.7 Research Methodology and Achieved Goals

This research converges multiple science divisions together. The combination of microelectronics, programming and cryptography and the effect of targeting one entity resulting in other entity of science is challenging.

This research can broadly be classified in two major segments.

1. Survey on Hardware Trojans, studying Parametric Hardware Trojans (PHTs) in detail with Intel's Ivy Bridge's microprocessor as a case study and possible prevention and countermeasure against PHTs.

2. Detection of trojans via Side Channel Analysis (SCA).

In second segment, a MATLAB based model is proposed for deteting PHTs using SCA. As for proof of concept, an FPGA implementation of trojan was also tested while analyzing the results obtained through the proposed detection mechanism. This step had three major steps:

1. Verilog coding,

2. MATLAB coding and

3. Real-time power trace acquisition via FPGA board.

1.7.1 Achieved Goals and Contribution of this research

The achieved goals and contribution of this research can b summarized as follows:

1. Proposed a comprehensive taxonomy based on different researches on identification of hardware trojans.

2. Proposed SCA model for detection of dopant level PHTs.

3. Analysed the effectiveness of SCA on FPGA implementation of dopant level PHT at HDL level.

4. Proposed countermeasures for the analysed PHT.

1.8 Thesis Organization

This thesis report has 7 chapters. Chapter 2 has literature review of hardware trojans and their taxonomy. This chapter also explains how parametric hardware trojans can be inserted. Based on the literature review, a taxonomy of hardware trojans is proposed. Chapter 3 is on a case study on a high profile, meaningful target of PHT i,e, Intel Ivy Bridges RNG. Chapter 4 is detailed analysis of detection of PHTs. Common detection techniques are discussed and their effectiveness for PHT is discussed. At the end of chapter, based on analysis, an SCA model for detection of dopant level PHT is proposed. Chapter 5 gives an experimental analysis of proposed SCA model. Real time acquisition of power traces is carried and related observations and limitations are documented.

Chapter 6 details the countermeasure for prevention and best practises that can avoid getting the infected hardware for critical projects followed by conclusion of dissertation detailed in Chapter 7.

Chapter 2

RELATED WORK AND LITERATURE REVIEW

2.1 Introduction

This chapter gives an insight of hardware trojans. First section deals with related work on hardware trojans. Then based on this literature review, a taxonomy is proposed The next section is dedicated to Parametric Hardware Trojans (PHT) which discusses its effects and other attributes in depth and in acordance with the proposed taxonomy. Next section details different methods for inserting PHTs which is followed by cryptographic attacks that are possible through PHTs. The conclusion of this chapter is discussed at the end.

2.2 Hardware Trojans

Hardware trojans are malicious manipulation on electronic circuit design on board. It is a very broad term and alot of spacs is present for an attacker to insert a trojan. The identification of these trojans is mostly based on their dominant characteristics. Based on this, hardware trojans can be classified according to their wide and varied attributes such as their activation mechanism, physical attributes, actions performed, and insertion phase of trojans etc (4) (5), (6), (7), (8).

2.2.1 Attributes of Hardware Trojans

It is necessary to classify trojans-under-study for ease of its understanding in depth. Various researchers have proposed taxonomy of hardware trojans. Different attributes according to which a hardware trojan can be identified are discussed below.

2.2.1.1 Insertion Phase

Depending on the technology, such as FPGA or ASIC, on which the trojan is inserted, insertion phase classifies the trojans on the basis of the phase of deploying an IC at which the trojan is inserted. Generally, IC development cycle is comprised of five phases as shown in figure 2.1. Threats at different phases of typical IC deployment are briefly explained below.

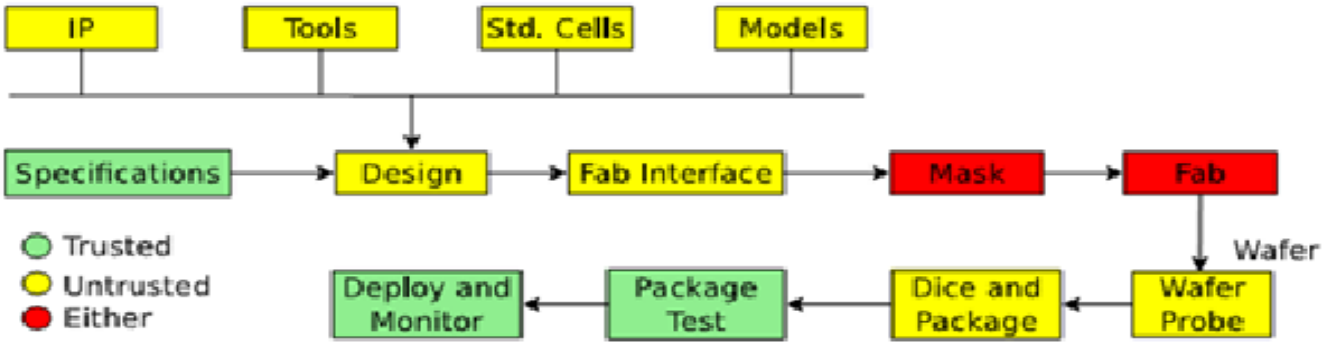


Figure 2.1: Vulnerable Phase of IC Deployment (Photo Courtesy: Chakraborty et a)

1. Specification phase: In this phase physical and functional characteristics of the system are mentioned. Hardware trojan at this level can be alteration of such specification such as change in its requirements. Mostly this phase is considered as the trusted phase of cycle as specifications are defined and delivered to the foundry in a trusted environment.

2. Design phase: The second in step are functional, logical, timing and physical constraints. Moreover standard cells and third part IP cores are also mapped in this phase. This is one of the most vulnerable module of the cycle as it compromises of multiple third party services. Malicious IP cores or standard cells can be used to set hardware trojans.

3. Fabrication/ Manufacturing: During this step mask is created and mapped on wafers. Small intentional or unintentional changes in masks or mapping can result in drastic changes. Change in chemical composition like doping concentration can result in behavioural changes at transistor level assisting stealthiest of hardware trojans. Trojans inserted at this level are also referred as MAPLE (MAnufacturing Process LEvel) trojans (9).

4. Testing: After cutting in wafers and then into die, ICs are tested against various test vectors. The basic motivation of such tests is to detect manufacturing faults. For an attacker to hide its trojan against such tests, he has to either bypass the tests or take control over the test vectors such that its trojan is not detected. To avoid such attacks, test vectors are usually kept secret and confidential.

5. Assembly and package: Dies are then finally connected with other components and assembled on a PCB. Inappropriate connection can result into malicious functionality such as leakage of information through unshielded wires by electromagnetic waves.

It is likely to say that almost all phases of IC development cycle are vulnerable to hardware trojans. An attacker, with his technical background, motivation and budget can launch a hardware trojan anywhere

2.2.1.2 Abstraction Level

This attribute defines trojan at the layers of an IC at which it can be inserted (4).

1. System level: This is the top most level where IC developers translate user requirements into hardware modules, interconnections and their communication protocols. Small changes in any of these specifications can alter the behaviour of an entire IC or trigger a Trojan.

2. Development environment: Different CAD tools are used for synthesis, simulation, verification and validation of IC design. Trojans or modifications in the software can bypass verification or validation tests and make an IC apparently trojan-free.

3. Register Transfer level: Functional modules are translated into registers, signal and boolean functions. A simple change, for example, in counter increment can reduce the cryptographic strength of chip.

4. Gate level: Registers and boolean functions are represented in form of logic gates at this level. A trojan at this level can be as simple as an additional XOR gate which can act as a comparator which will continuously monitor the signals.

5. Transistor level: Logic gates are made of transistors. Trojan designers can have control over power and timings of gates here. Different modifications on transistor's parameters can result in different kind of output that a trojan designer can use.

6. Physical level: This is the design's physical level where the location, distance and size of all the components and their interconnections are described. Just by increasing the distance between two gates, for example, can cause reasonable delay in gate's functionality.

2.2.1.3 Activation Mechanism

With respect to activation mechanism, trojans can be classified as ones that are always activated or those who always require some triggering mechanism to activate. Depending on their activation mechanism, trojans can be classified as follows:

1. Externally activated: Hardware components that can interact with the external environment, when manipulated, helps in triggering the Trojan. For example they can either be triggered

through user input from some sensor like keyboard or an output from some hardware component like Antenna. The Trojan can either stay at this state forever or return to its dormant state after specific time or when the condition is fulfilled.

2. Internally activated: Unlike externally activated Trojans, these Trojans rely on the triggering condition or event that resides inside the device or on an IC. Generally internally activated Trojans can either be:

   (a) Always on.

   (b) Triggered: Triggering mechanism of internally activated Trojans is different from that of externally triggered Trojans.

      i. Time based: They can rely on the clock of the system or the counter after which the Trojan is activated.

      ii. Physical condition: the other way is to rely on the physical condition of the device and its parameters such as temperature.

2.2.1.4 Physical Characteristics

This category of taxonomy classifies according to the physical changes that are made on the device to construct the trojan. Two categories are:

1. Functional class: Devices/ICs on which trojans are inserted with the help of an additional gate/transistor or a component falls in this category. Additional components are added in such way that they do the same functionality but gives extended capabilities side by side to the attacker.

2. Parametric class: In this class trojans are inserted by affecting parametric properties such as clock or timing parameters and power usage. This is achieved by directly influencing the intrinsic IC properties including that of wire and transistor geometries. No additional gate is added and changes are made on present design of IC.

2.2.1.5 Effects

HTs can be classified according to theri effect on device after trojan activation. Some of the major effects are:

1. Functional Change: Some trojans result into either modifying or disabling the functionality of the target device and cause subtle changes. This directly compromises integrity of an information system that is also difficult to detect. For example, a trojan can cause error detection module to accept inputs that should be rejected normally. Such functionality modification are limitless; the actions resulting from this class are only constrained by resources, imagination, and skill of an adversary.

2. Change in Specification: This class of hardware trojan can perform variety of actions. Examples include limiting the processing capability of a system by modifying system clock, or by replacing computational or I/O units that are functionally equivalent but have reduced throughput performance.

3. Information Leakage: This is basic and main threat to any crypt-devices. Sensitive information can be leaked through covert or overt channels such as radio frequency, optical or thermal power, timing side channels, and interfaces such as RS-232 and JTAG (Joint Test Action Group).

4. Downgrading Performance: A Trojan can have as simple effect as that of downgrading performance by intentionally changing device functional, interface, or parametric characteristics such as power and delay [33].

5. Denial-of-service (DoS) trojans prevent operation of a function or resource, exhausting scarce resources like bandwidth, computation, and battery power. This simple Trojan can have simplest to the most deadly effect like physically destroying, disabling, or altering the devices configuration.

2.2.1.6 Location

A hardware trojan can be inserted at different locations in a system (4). The location of trojan influences the complexity of design, difficulty in its insertion as well as actions. COmmon insertion locations are:

1. Processor: These are trojans that are embedded into the logic units of processor. A trojan in processor might, for example, change the instructions or execution order.

2. Memory: Trojans in memory blocks and their interface units fall in this category.

3. I/O: Trojans can reside in a chips peripherals. These peripherals interface with the external components and can give trojans control over data communication between the processor and the systems external components.

4. Power Supply: Trojans targeting this can alter voltage and current supplied to the chip, causing failures.

5. Clock: Trojans in clock grid can change clocks frequency and insert glitches in clock supplied to chip. These trojans can also freeze the clock signal supplied to chips functional

modules

2.2.1.7 Cost

This classify fault injection attacks into low-cost ones (which a single attacker with a modest budget) and high-cost ones (requiring highly skilled attackers with a large budget).

1. Low Cost: This cost is well within the means of a single motivated attacker. Under-powering of device, injection of well-timed power spikes or temporary brown-outs into the supply line of circuit, tampering clock signal, altering environmental conditions such as temperature and tapping through EM interface all are included in low cost fault attacks.

2. High Cost: This class is defined as attacks that need high technical skills and high budget. Moreover it requires direct access to the silicon die. Resultant attacks are very precise and stealthy in detection.

2.3 Proposed Taxonomy

Based on different attributes, discussed in section 2.2.1, we have proposed a comprehensive taxonomy in figure 2.2. This figure comprehends different attributes of typical hardware trojans. It is adopted from various surveys and is a generic, comprehensive taxonomy covering almost all kind of hardware trojans.

2.4 Parametric Hardware Trojans

According to figure 2.2, the trojan that is studied in this research is termed as Parametric Hardware Trojans (PHTs). This is because the most dominant feature of these kinds of trojans is their stealthy detection. The description of this trojan is described in terms of the proposed taxonomy.

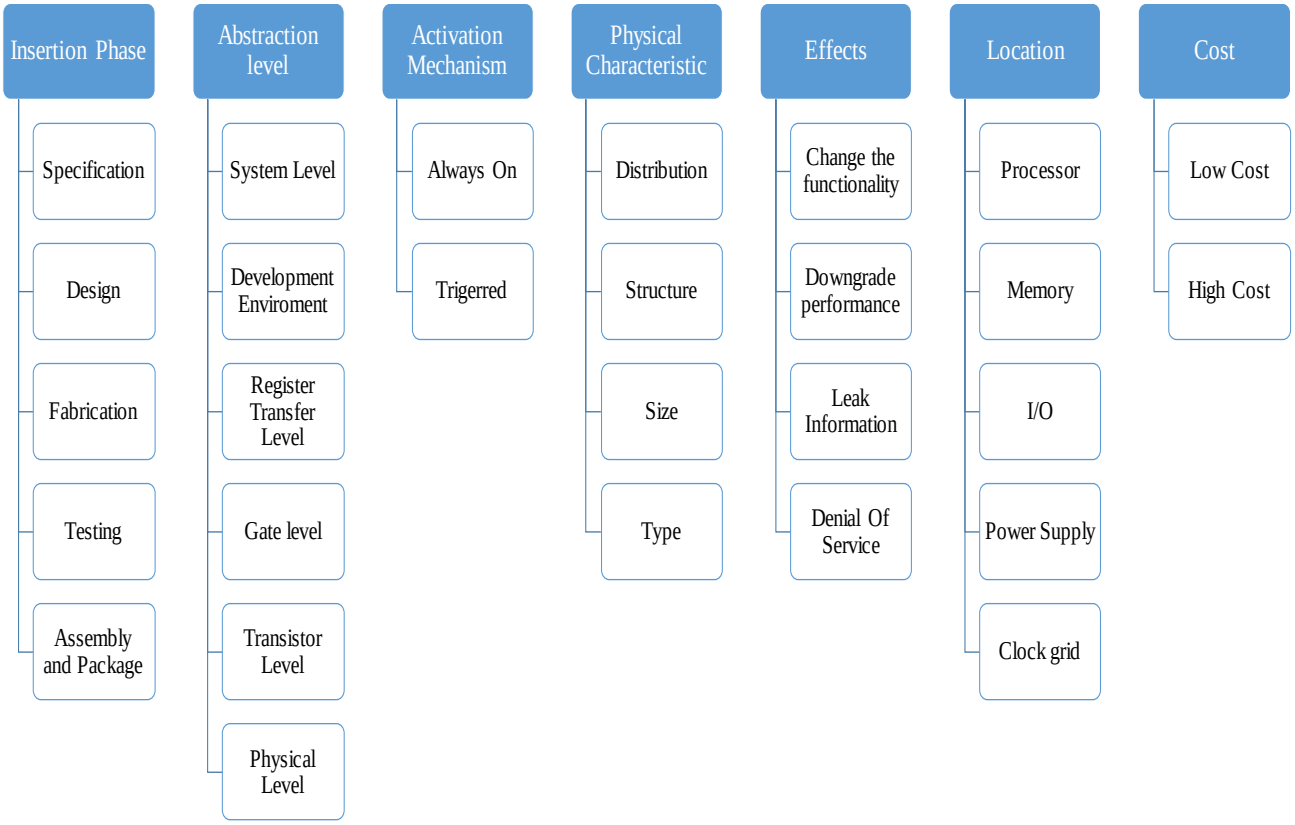


Figure 2.2: Taxonomy of Hardware Trojans

1. According to first attribute of figure 2.2, these Trojans are inserted at Manufacturing Phase of ICs. For that they are often referred as MAnufacturing Process LEvel (MAPLE) trojans.

2. At 'abstraction level',the second attribute, PHTs are based on manipulating the logic of gates. This is done by changing their electronic properties at gate level, such as changing VTC of gates, in such a way that the metal layers above them are untouched. Hence they lie in 'Gate-Level@ of abstraction level.

3. Acording to different 'activation mechanism', PHT can be internally or externally activated. Different possibilities of inserting PHTs are already described in section 2.2.1.3.

4. Fourth attribute of 2.2 is 'Physical Characteristics'. PHTs make changes on already available IC design such as changes in parameters of existing wires and logic. This is the most dominant feature because of which they are named as Parametric Hardware Trojans. These trojans are only possible when the subjected chip is well understood and the attacker exactly knows what she wants. PHTs have destructive impact on chip because the changes made for inserting trojans are irreversible. They are therefore termed as invasive attacks on chip.

5. The 'effect' of such trojans can vary from functional changes to leaking information, not only through side channel attack but by carefully inserted backdoors. Other types downgrade performance of the targeted device and thus reducing their reliability. The actions resulting from this class of trojan are only constrained by the resources, imagination, and skill of an adversary.

6. PHTs can reside anywhere on an IC as discussed in figure 2.2.1.6. Trojan at different locations result in different effects.

7. It is not possible to have an access of silicon die and to make changes on it without sufficient technical expertise. Methods of changing parametric properties of silicon die requires highly sophisticated equipment such as FIR or NIB. These changes includes tampering through laser beams such as Near-InfraRed (NIR) lasers, that can also radiate on silicon die from back, and focused ion beam (FIB) that enables an attacker to arbitrarily modify the structure of a circuit, reconstruct missing buses, cut existing wires, mill through layers, and rebuild them in the most precise and accurate manner. Therefore this class comes under high-cost fault injection techniques in last attribute of proposed taxonomy.

2.5 Methods of Inserting PHTs

For inserting PHTs, sophisticated techniques are required. There are many ways to insert PHTs and it totally depend on available resources, levle of sophistication required and avaialble budget. The most common PHT's insertion techniques are discussed in this section (10) (6).

2.5.1 Supply Voltage

Variations in Supply Voltage during execution may cause a processor to misinterpret or skip instructions. This method is widely researched and practised by the smart-card industry but does not often appear in open literature. There are two main ways of altering the supply voltage

1. Under-Powering: When chip runs with depleted power supply, then it is possible to insert transient faults starting from single bit errors and becoming more invasive as the supply voltage gets lower. Since this technique does not require precise timing, faults tend to occur uniformly throughout the computation, thus requiring the attacker to be able to discard results that are not fit to lead an attack

2. Well timed power spikes: High variations in a power supply can also affect the device performance such as causing a processor to skip or misinterpret an instruction and other memory faults. More accurate the voltage drop is, the more accurate power spike would be. The level of difficulty increases with the increase in clock rate.

Both techniques require an attacker to access the power supply line of IC.

2.5.2 Clock

Power glitches actually affects clock signal. Shortening the pulses, corrupts the signal that causes errors in a stored byte or multiple bytes. Direct access to power supplies is needed just as in power supply disruption Trojans in this case.

2.5.3 Environmental Conditions

ICs are made to work properly at particular temperature limits. Exposure to too high or too low temperatures can induce faults. Using ICs in non-suitable temperature can alter data stored in memory. In extreme cases invasive faults in sensitive devices can occur or the device is destroyed. This sensitivity against temperature can be exploited.

2.5.4 Electro-Magnetism Waves

EM disturbances is an easiest and excellent way to tap in the computation of encapsulated device. This is the result of eddy currents induced in the circuit which can be recorded.

2.5.5 Optical Faults

Trojans inserted through EM disturbances are not as precise as by EM waves since they disperse through the layers of IC and cannot penetrate in depth. For accuracy, the device is decapsulated and advance optical fault injection techniques are used. As semiconductors and conductors are inherently sensitive to laser ionization, it is possible to cause as precise errors as switching of transistors.

1. X-rays and ion beams can also be used as fault sources (although less common). These have advantage of allowing the implementation of fault attacks without necessarily de-packaging

the chip.

2. Optical fault injection technique is constantly advancing, helping to make changes on board more precise and accurate. The light spot size on a chip die is shrinking and is now physically limited by wavelength of the photons. Using focused laser beam, a single bit in memory can be set or reset - Giving an advantage of penetrating the chip from front as well as backside.

   Two of the finest technology are Near-infrared (NIR) lasers and Focused Ion Beam (FIB) lasers. The latter one can modify the structure of a circuit, reconstruct missing buses, cut existing wires, mill through layers, and rebuild them.

2.6 Cryptographic Attacks by PHTs

PHT assists active-invasive attacks on the crypto-processors. Active attacks in a way that they require direct unsupervised access to the device. One can manifest in various ways from such trojans. Cryptographic PHT requires high technical background and sufficient knowledge of the chip. There are mainly three PHT attacks that are reported in literature till date, one affecting IC process reliability (11) and other two affecting the cryptographic security. The PHT based crypto-attacks in literature are discussed in next section.

2.6.1 Stealthy Dopant-Level Hardware Trojan

In 2012, George T. Becker introduced sub-transistor level hardware trojan that needs modification at dopant mask only (1). This trojan bypasses most of the post-manufacturing trojan detection mechanism. Its effect are studied at the NIST verified RNG recommended by Intel. In this case study AES at counter mode is subjected to the attack and the functional test of RNG by Intel is also bypassed. Second case study shows effects of this trojans on hidden side-channel. The

modification is done on power profiles of two logic gates. Only the attacker can manifest from this trojan and can get the keys using the side-channel attacks.

2.6.2 Parametric Trojans for Fault-Injection Attack

Another trojan presented in (9), proposed in 2014 is modification of the previous attack. It assumes that provided the functional test not bypassed, the above attack can be detected through functional tests because of deterministic changes on the gate. Unlike previous trojan, this trojan does not stay active all the time. It is a multi-stage trojan requiring 2 or 3 fault injections. The proposed trojan is triggered in presence of slightly reduced supply voltage. Under such circumstances the trojan inverter will flip its state. As a case study, PRINCE cipher is subjected on which differential fault analysis is performed. The fault-affected output values are collected, and differential cryptanalysis is used to derive the secret key.

2.7 Conclusion

This chapter was a survey of hardware trojans but was focused on one class of it known as Parametric Hardware Trojans. Different ways of inserting PHTs gives different effects which are selected by the attacker and depends on his motivation. Two major cryptographic attacks are discussed which are till date present in literature.

Chapter 3

CASE STUDY: INTEL'S RANDOM NUMBER GENERATOR

3.1 Introduction

In this dissertation, the danger and effectiveness of PHTs is studied in a high profile target that is Intel's cryptographically secure Digital Random Number Generator (DRNG). The reason for choosing this target is because of its potential for real-world impact. It has been rigorously tested by independent organizations. Due to its importance, enormous interest of cybetsecurity community can be seen on this subject. This chapter first discusses the internal working Intel's Ivy Bridge RNG and in next section the dopant level PHT on this RNG is explained in detail.

3.2 Intels Ivy Bridge Digital Random Number Generator

Digital Random Number Generator (DRNG) used in Ivy Bridge is NIST SP800-90, FIPS 140-2, and ANSI X9.82 compliant and is rigorously tested by an independent security company (12). It is used in the Ivy Bridge processors but will most likely be used in many more designs in the future. Currently its products are branded as 3rd Generation Intel Core processors for client systems, and Intel Xeon v2 Processors for server systems.

3.2.1 Applications of RNGs

Applications of RNGs are in different areas such as gambling, statistical sampling, computer simulation, cryptography, completely randomized design etc. All these domains demands an unpredictable output. In cryptography, these generators are employed to produce secret keys such as

generating public/private key pairs for asymmetric (public key) algorithms including RSA, DSA, and Diffie-Hellman. Keys for symmetric and hybrid cryptosystems are also generated randomly. Other then key generation, DSA and ECDSA digital signature standards require a random value when each signature is produced. Moreover RNGs are also used to encrypt messages or to mask the content of certain protocols by combining the content with a random sequence. They are used to create challenges, nonces (salts), padding bytes, and blinding values. Flaws or attacks on challenges can result as serious as compromised OpenSSL keys including server certificates, SSH login keys and email signing/encryption keys, common prime factor in unexpectedly large number of RSA moduli.

3.3 Internal Working of Ivy Bridge RNG

This section gives a brief understanding of internal working of Intel's RNG. It also explains AES-128 and its counter mode comprehensively.

Intel RNG consist of an Entropy Source (ES) and a digital post processing logic. The entropy source is an asynchronous metastable circuit that generates random bits at 3GHz rate approximately while the digital post-processing consist of Online Health Test (OHT) and Deterministic Random Bit Generator. (DRBG). The function of OHT is to monitor the random numbers generated by ES through statistical tests on each of the 256-bit output of ES. This is required to maintain minimum entropy. Other parts of post-processing is DRBG that is designed to provide cryptographic security for final random numbers generated. DRBG is composed of two parts, conditioner and rate-matcher. Both parts are based on AES. The conditioners main purpose is to cryptographically generate seeds for rate-matcher. And it does so by processing OHTs output, stored in Online Self Tested Entropy (OSTE) through AES. The lower and upper part of the conditioning pool are updated separately. The sequence of these different modules can be seen in

figure 3.1.

The purpose of this mechanism is to introduce full randomization. After the conditioning pool is

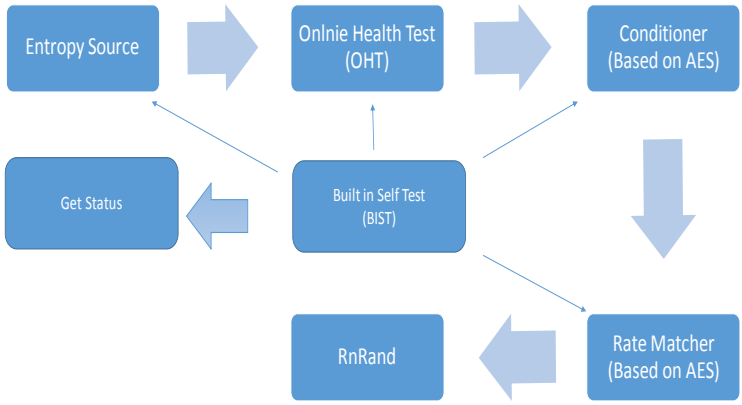


Figure 3.1: Block Diagram of Ivy Bridge RNG

updated, it reseeds the internal states of rate-matcher. The lower half of conditioning pool updates Key **k** and the upper half updates counter **c**. The pseudocode for reseeding, (c, k) = Reseed (s, t, c, k) where **s** and **t** are the two halves of the conditioning pool, is given in equation 3.1.

$$\begin{aligned} c &= c + 1, x = AES_k(c), \\ c &= c + 1, y = AES_k(c), \\ k &= k \oplus x \oplus s, \\ c &= c \oplus y \oplus t. \end{aligned} \tag{3.1}$$

The behaviour of rate matcher can be summed by the function (r, c, k) = Generate (c,k). Given the counter value and key, rate-matcher will generate 128 random bits **r** and updates the state registers

in equation 3.2.

$$
\begin{aligned}
c = c + 1, r = AES_k(c) \\
c = c + 1, x = AES_k(c) \\
c = c + 1, y = AES_k(c) \\
k = kx \\
c = c \oplus y
\end{aligned}
\tag{3.2}
$$

A rate matcher needs a new seed after generation of 512 128-bit random numbers (65536 bits total). So after 512 random number are generated, reseed function will be called again. Internal working of AES can also be visualised through figure 3.2.

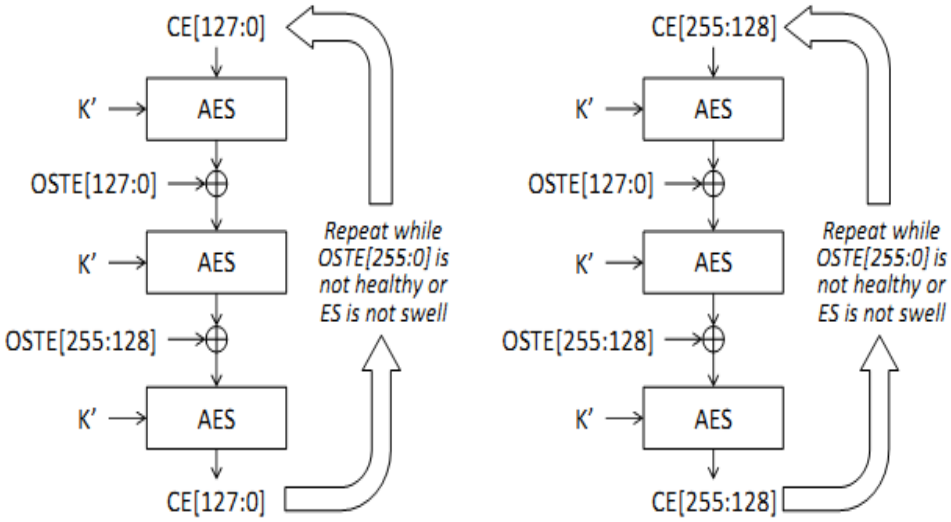


Figure 3.2: AES-128 in RNG (Photo Courtesy: Mike Hamburg)

Short description of AES-128 and its counter mode is explained in next subsections.

3.3.1 AES-128 Encryption

Advanced Encryption Standard AES 128 is one of the three symmetric encryption block ciphers of AES formerly named as Rijndael cipher (13). The other two versions of AES are AES-192

and AES 256. Unlike DES, AES follows a substitution-permutation network. AES-128 processes data blocks as 44 column matrix known as *state*. 128 bit state as plaintext, cipher key and cipher-text. Plaintext is processed through 10 rounds, each having similar 4 operations. All these rounds use a round-key which is derived from the cipher-key. The last round, 10th round, generates ciphertext as its output. General working of AES encryption is shown in figure 3.3.

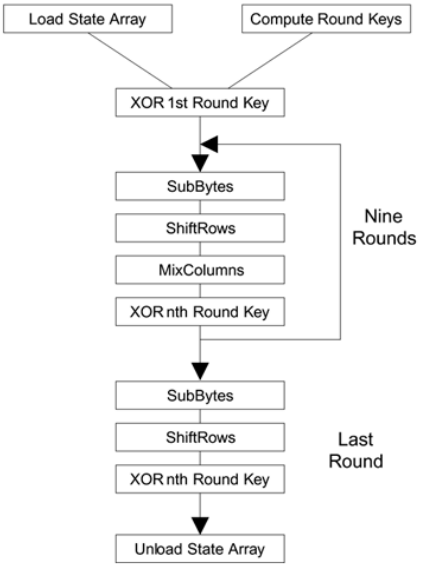


Figure 3.3: AES Encryption, (Photo Courtesy: NIST Special Publication 800-38C)

In first step, ten round-keys are generated through the cipher key by a key schedule. Each of these keys are used in respective round of encryption. First round, i.e. initial round, xors bit-wise plaintext with the first round key. After that, a four stage round is repeated nine times. In first stage, each byte from input state matrix is replaced with another byte from the lookup table known as *Sboxes*. In second stage, this intermediate state is processed through ShiftRows. This is a row-operation-function in which each row is cyclically shifted by a certain offset. This offset is equal to the number of the row on which the operation is applied. Third stage does the MixedColumn operation which is an invertible linear transformation. It is a column operation. Together with

ShiftRows, MixColumn operation provides sufficient diffusion. The last stage is bitwise xoring of this intermediate state with the RounfKey. In the final round, 10th round, the MixColumn operation is skipped.

3.3.2 Counter (CTR) mode

Counter (CTR) mode in one among the five modes-of-operation of block ciphers. The other modes are Electronic CodeBook (ECB), Cipher block Chaining (CBC), Cipher FeedBack (CFB) and Output FeedBack (OFB) (14). CTR mode is a confidentiality mode. It generates a keystream by encrypting a random counter value which is then xored with the plaintext. Counter values should have the property of being unique within the complete encryption process and non-repetitivly for sufficient duration of time. Counter values are generated by various schemes. In this research, RNG that is one of the way to generate counter values, is considered.
Let the counter values be defined as **(T1, T2, T3.. Tn), O** is the keystream, **P** is plaintext and **C** is cipher text, CTR encryption is defined in equation 3.3.

$$\begin{aligned} O_j &= Cipherk(T_j) where j = 1, 2, 3n \\ P_j &= P_j \oplus O_j \\ C_n &= P_n \oplus MSB(O_n) \end{aligned} \tag{3.3}$$

To encrypt a 128 bit plaintext, 128 bit counter is encrypted through AES-128 and the output is xored with the plaintext to produce a 128 bit state. The ciphertext is final xoring of the plaintext with the most significant bit of encrypted counter.

3.3.3 Built In Self-Test

To balance the trade-off between performance and security, Ivy Bridges RNG has only one test, Built-In-Self-Test (BIST) to check for its correct functionality. It is a two phase test. The first phase tests the logic and health test while the second test checks ES and initialization of DRBG. For first phase, the ES is disconnected and replaced with a Linear Feedback Shift Register (LFSR). The DRBG is initialized to generate random numbers and a 32 bit CRC is calculated. This value is matched against hard-wired CRC which upon matching verify the circuit logic.
In second phase, the ES is connected again and is run for the normal operation. When ES swells, the samples are processed after replacing all zero key and counter state to generate random number, in this way the ES is also teste. For the actual DRBG process to start, BIST has to pass otherwise RNG will give all zero output with a clear flag indicating that there is a problem.

3.4 Dopant Trojan for Intels DRNG

In this section dopant trojan for Intel DRBG as given by Georg T. Becker in (1) is explained in detail. The random numbers of DRNG is the output of the function AES(c,k). Here AES computes on 128 bit data block known as state. The complexity is therefore dependent on 128 bit secret key and 128 bit unknown counter that is reseeded by the well protected conditioner pool. Thus the attacker has complexity of 128 bit to correctly guess the unknown. In order to reduce the complexity, there should be some means to control few bits of the unknown such a way that only the attacker knows it. The goal of this stealthy trojan is also same,it manipulates the counter bits in such a way that they are fixed and known to the attacker but are unknown to the third person. This is done by carefully manipulating the flip-flops of the internal state of rate-matcher in such a way that they are fixed. The location of trojan insertion in complete RNG process can be seen in

figure 3.4.

In rate matcher of RBG, the first step of the attack is to fix the register containing key. Now

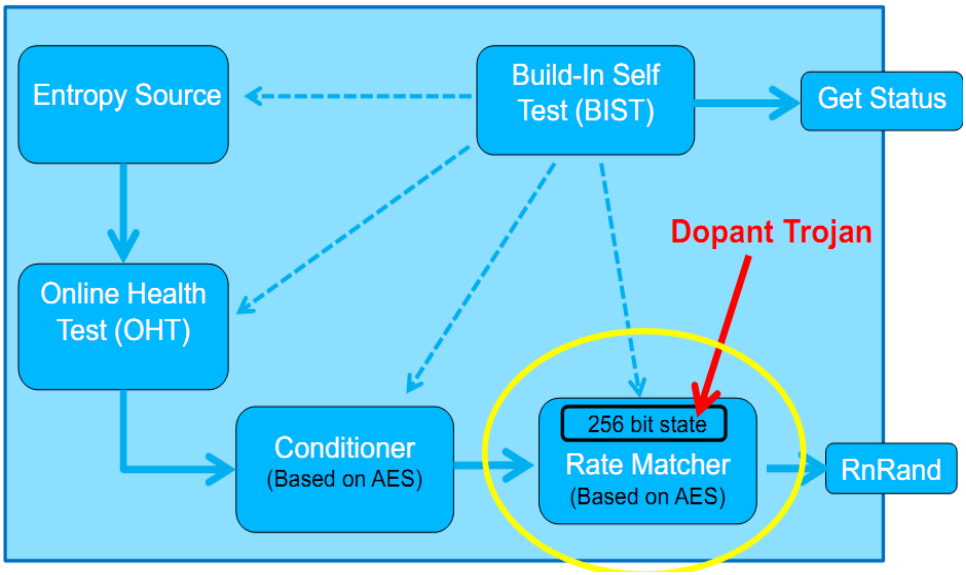


Figure 3.4: Trojan inserted in Intel Ivy Bridge RNG

only 128 bit counter bits are unknown. The complete register of counter can also be fixed by same dopant trojan but the aim here is to successfully pass NIST RNG test as well and that the output of RNG should be random for third user. For this reason only **x** number of bits of 128 bit counter are fixed, leaving behind 128-x=n bits unchanged. Now the complexity of the entire RNG is dependent on only **n** bits. On the other hand, RNG behaves perfectly randomly for the third person. Fixed register of the 128 bit counter can be visualised as shown in figure 3.5.

3.4.1 Defeating BIST

Same dopant Trojan is used on BIST. But to make the Trojan stealthy, the attacker will not like to make the BIST pass for every input. General process is shown in figure 3.6

The main challenge for bypassing BIST is to bypass first phase of the test where CRC is calculated and tested against known, hard-wired CRC. The calculated CRC is of the last four output stored

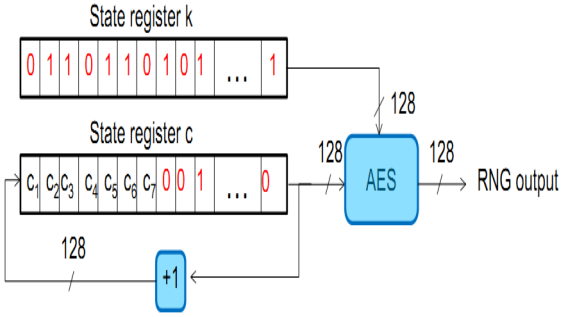


Figure 3.5: Trojan in AES-128 counter (Photo Courtesy: Georg T. Becker)

in the buffer. The attacker needs $2^{32}/2$ tries on average to guess the two, 32 bit state register, **c** and **k**, so that it can compute the known CRC. This can be done through simulations by trying different values of **c** and **k**. This trojan also bypasses the statistical tests designed to check for

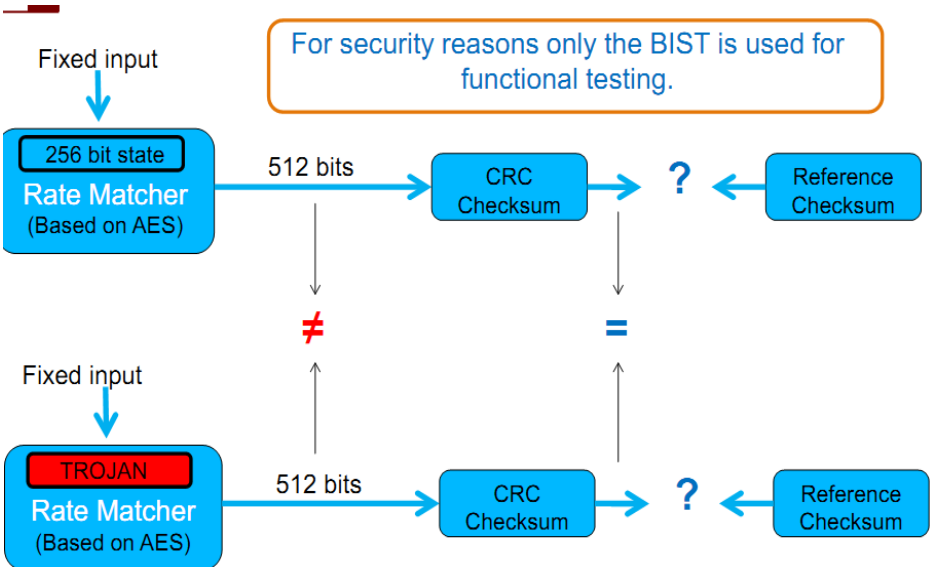


Figure 3.6: Trojan in BIST (Photo Courtesy: Georg T. Becker)

RNGs to be NIST SP800-90 and FIPS 140-2 compliant. The next chapter will focus on how such Trojans can be detected through side channel analysis.

3.5 Conclusion

A dopant level trojan introduced by George T. Becker was the topic of this chapter. Intel's Ivy Bridge RNG is taken as a targeted device. Inner working of RNG is explained and a PHT is inserted at its AES module of rate matcher. The effect of this module is discussed and it can be visualized how bit level changes compromises the complete well and rigorously tested designed structure.

Chapter 4

DETECTION TECHNIQUES FOR DOPANT-LEVEL PARAMETRIC HARDWARE TROJANS

4.1 Introduction

This chapter is dedicated to detection of PHTs. Although some of these techniques are used to detect other kind of hardware trojans too, but those which are found effective in detecting PHTs are discussed in detail. This chapter also comprehends already research carried out for detection of trojans. A short conclusion on this related work and literature review is mentioned at the end.

4.2 Overview of Trojan Detection

Different detection schemes target different trojans. Therefore there is a wide variety of detection mechanism as well. Mostly trojans are introduced by using additional circuit on board which are detected either by optical reverse engineering or by IC finger printing (15). The former technique focuses on the apparent changes made on ICs which are mapped against a golden chip, a chip considered as flawless. The latter one takes power or other side channel's patterns which are mapped against the golden chips pattern. This mapping methodology is refined and proposed by many authors. Most current research focuses on detecting trojans at post-fabrication level because fabrication process is currently seen as the weakest link in the IC development cycle.

PHTs are sophisticated and so far the stealthiest emergence of trojans. The key characteristic of these manufacturing-level trojans are their ultra-low detectability by all known means: func-

tional testing, side-channel analysis, and visual inspection. Therefore they require advance and sophisticated detection mechanisms.

4.3 Methods of Detecting PHTs

Since PHTs are inserted almost at the last stage of IC manufacturing, they bypass most of the fault-detection tests. our focus is to study post manufactuing detection techniques because we are considering to detect trojans after ICs have been delivered to the intended consumer.

As shown in figure 4.1, there are three main post-manufacturing detection techniques. In this figure they are classified under the category of destructive and non-destructive means of inspecting ICs. Destructive techniques involve those methods that need to evaluate ICs in such a way that they are no more of use after the evaluation. These techniques includes removing polysilicon layer, wires etc. Usually these methods are done only with few ICs from a lot to randomly check for the intended quality. If the selected chip comes up to the ideal results, it is marked as golden chip. Such golden chips are later on used for inspecting other ICs with non-destructive means such as comparing visual or side channel results.

Non Destructive detection techniques are further categorized into techniques where golden chips are required for verification and those where golden chips are not required.

In next sections, we describe the general methodology and related work on the detection techniques listed in figure 4.1.

4.3.1 Visual Inspection

Passive Voltage Contrast (PVC) is a visual comparison test for detecting failure localization in ICs. It is brightness difference in the images taken by FIB (Focused Ion Beam) or (Scanning Electron Microscope) SEM. The brightness difference is between floating structures and grounded

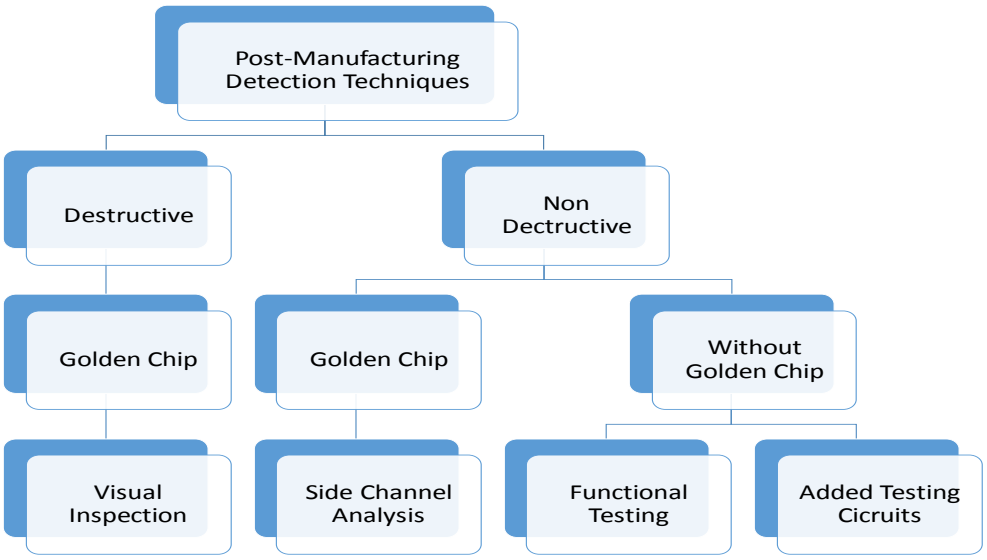


Figure 4.1: Post Manufacturing Detection Techniques (Photo Courtesy: Dusko Karaklaji)

structures. Under PVC test, where no external power source is attached and the die is partially decomposed, floating structures should be bright under the influence of primary beam of FIB or SEM. Similarly, the grounded structures should appear darker in images. If there is any abnormality in the brightness of floating structures and darkness of the grounded structures then that localise a fault injection.

The wavelength used by FIB or SEM is very small that is why PVC is a successful way of detecting stealth hardware Trojans. All four possible dopant-well combinations are distinguishable with SEM through this procedure (16).

In this technique a primary beam which can be of FIB or SEM is injected onto a sample surface through its magnetic coils. The beam of SEM contains electrons while of FIB are ions. This beam causes secondary electrons from the sample surface to emit. These emitted electrons are contained in detector and are measured. Iteration of this measurement completes the contrast image. Figure 4.2 describes the general methodology involved in PVC.

PVC is an invasive detection technique, requiring stripping off IC layers and taking images of

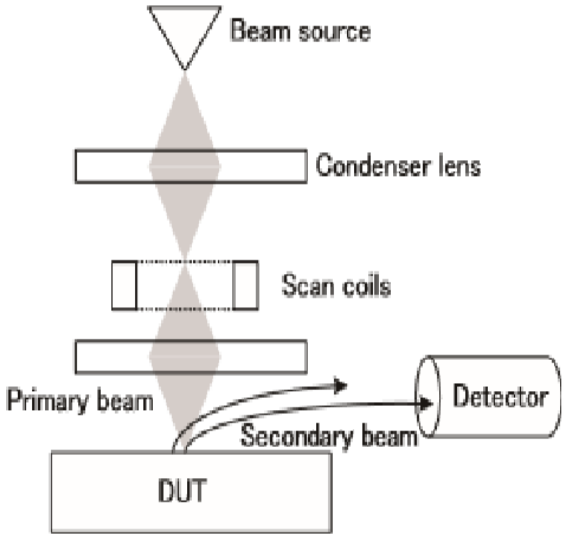


Figure 4.2: SEM/FIB measurement System (Photo Courtesy: Takeshi Sugawara

surface voltage by SEM or FIB, the same technology which is used for minor surgeries of ICs. Cost of this technique is directly proportional to the number of images taken. Therefore accurate results are costly but less time consuming. Visibility of PVC is more accurate in FIB as compared to SEM.

Meanwhile, (9) also gives a hint that a dopant modification cannot be detected by PVC if moderations are limited to regions not connected to contact plugs. Another technique, Active Voltage Contrast AVC is similar to PVC but requires external power or signal source to be connected. It offers more localization possibilities. The main advantage of AVC is the fact that, not just opens and shorts can be detected, but also it can detect failures caused by an increased resistance of conductors or contacts. This is very important when detecting stealthy hardware trojans that are introduced by fault injections by shortening the wells of gates. AVC is a non-destructive technique as it requires de-capsulation but not de-processing of the IC. Further comparison of AVC and PVC and whether to use FIB or SEM can be studied in detail (16).

4.3.2 Side Channel Analysis

Side Channel Analysis is a wide paradigm for detecting hardware trojans. It is dependent on physical implementation of the cryptosystem which can be exploited through different side channels such as power emitted or power consumed, electromagnetic waves, timing, sound, thermal or any other by-product of actual crypto-process, shown in figure 4.3. SCA is non-invasive detection technique. In case of PHTs, the manipulated logic gates have a parametric response on side channels that is different from regular gate's behaviour .

Using SCA for detection of trojan ICs, the problem of trojan detection reduces to detecting a trojan-signal hiding in IC process noise. This change in signal pattern is assumed to be differentiated through different comparison methods, using different parametric attributes such as path delay, leakage or dynamic current or power patterns of IC. Detection of these trojans get complicated with the decrease in size of the technology and trojan.

The concept of SCA using consumed power was first introduced by Kocher in 1999 (17). Like other side channels, the power consumed by IC is directly related to the operations it is performing and data it is processing. Power analysis require detailed knowledge of implementation of the cryptographic algorithm.

The first step of power analysis is to measure the power patterns while it is performing operations on a given data. These power patterns are called *power traces* of the cryptodevice. The second step divides the SCA into two categories: Simple Power Analysis and Differential Power Analysis. Both these techniques generally aims to retrieve the key. However, same concepts can be borrowed for detection purpose also. The details of these two techniques is discussed in this section.

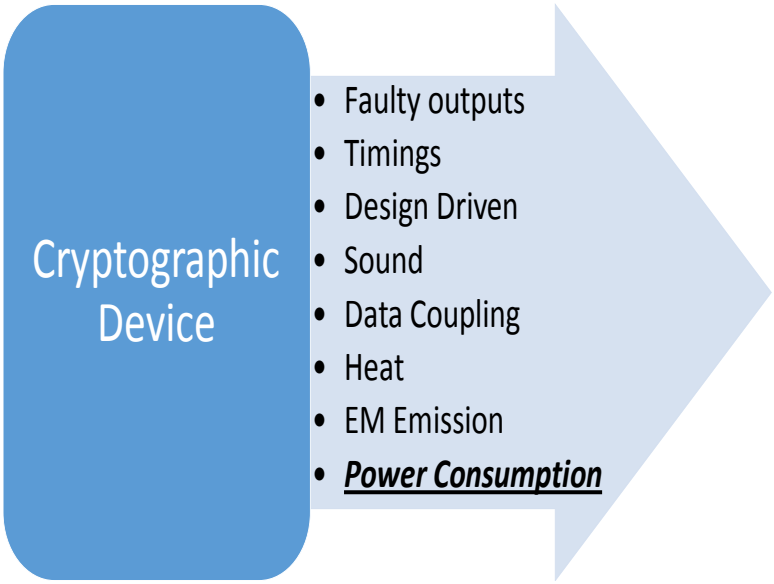


Figure 4.3: Side channels of a typical crypto device

4.3.2.1 Simple Power Analysis

Simple Power Analysis (SPA) is the simplest type of side channel analysis. In case of less process variation, SPA is an effective and easiest way to get the information about the process through side channels. There are two type of SPA; single-shot SPA and multiple-shot SPA. Single-shot SPA processes one trace and does the SPA on it while multi-shot take more than one or few trace against the same data and then perform SPA.

General working of SPA involves visual inspection of the power trace. Since the power consumed is dependent on the number of resources used by the instructions and the value of the data it is processing. The difference of the traces while performing operations with and without a trojan can therefore leak information.

SPA is only practical when sufficient knowledge about the internal structure of IC and the running algorithm is known. Moreover data dependant power consumption should be visible from the trace that is to say the process noise should be minimum. This is usually not the case because the

power trace has both intentional and unintentional noise added in it. It therefore gets difficult to spot the SPA leaks even if they are present in the power trace.

To overcome this situation Differential Power analysis becomes handy. This second type of SCA exploits the statistical relation between the data and operation dependent traces. More advance techniques such as Collision attack, Algebraic attacks and lattice based methods can be used to refine results obtained from SPA (18).

4.3.2.2 Differential Power Analysis

Differential Power Analysis (DPA) is an advance form of SCA. It is more efficient and robust way of retrieving the key as compared to SPA. It also overcomes the problem of noise and other process variations by statistical analysis and efficiently exploits the data-dependent correlation between the signals. There are 5 major steps for retrieving a key through DPA (17; 18). They are listed below:

1. **Power trace acquisition** The first and most critical step is power trace acquisition. They are acquired during the encryption or decryption and sampled to create corresponding trace matrices. Other signal processing i.e. eliminating noise and averaging the signals is also carried out in this step.

2. **Intermediate result** In second step, a selection function **f(k,d)**is selected that belongs to the crypto-algorithm under attack with key **k** and available variable data **d**. For retrieving the key, all possible values of **k** are considered and the resultant values from the selection function are evaluated.

3. **Mapping Intermediate values to power consumption** The hypothetical intermediate values, evaluated in the previous step are mapped against hypothetical power consumption

that are derived through different power models and a resultant matrix is created. The most common power models are Hamming distance Model or hamming weight model (18).

4. **Comparison of two power matrices** The last step involves the comparison of two matrices through statistical analysis. The resultant matrix contains the comparison values. Highest value in the matrix show the highest correlation between two matrices. In acquired power traces, these high values are represented by high peaks as shown in 4.4. The indices of these high values are traced back to the matrix that will be the actual key.

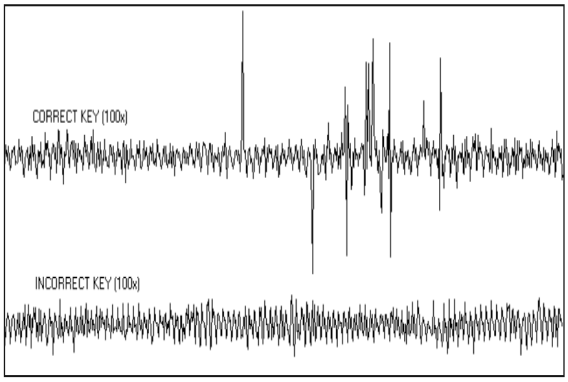


Figure 4.4: Successful DPA by Stefan Mangard

The number of traces required to successfully retrieve key from power traces is dependent on signal to power ratio of the power signal.

4.3.2.3 Advanced Differential Power Analysis

IC Fingerprints Agrawal et al. gave a formal method of detecting hardware trojans in complex scenarios (15). He created a simple yet destructive trojan that could insert a fault in Chinese Remainder Theorem (CRT) inversion step for RSA signature computation which, resultantly, compromised the RSA key.

The detection of trojan was based on comparing **IC fingerprints** which can be power, temperature or EM profiles. For that, two batches of ICs were created. One batch had small number of chips which were invasively tested with techniques such as demasking, delayering and layer-by-layer comparison of X-ray scans with the original design specification. ICs that passes through all the tests are names as *golden chips*. This step is likely to be expensive, but since it is only done for a few selected ICs from an entire family, the cost when amortized over all the ICs, may still be acceptable. The golden chips were run over different I/O tests to collect sufficient data on their behaviour against these different inputs. The criteria of evaluating the behaviour can vary, depending on the selected side channel. In this case, power signals are used. The power pattern of golden chips were then later used as fingerprints.

The second batch has rest of the ICs to be tested which are statistically verified by comparing through the fingerprints of golden chips with the fingerprints against same side-channel. The trojan detection problem is then considered as signal characterization problem.

Statistical approach for detecting hardware trojans that is based on the analysis of an ICs power supply transient signals is analysed in (19). Four different **signal calibration** techniques are analysed to determine their relative effectiveness on reducing the adverse effects of process and test environment variations.

A comparative study for detecting trojans in three cases, one with no trojan (denoted by *no-trojan*), second with one extra gate (denoted by *1gate*), and third with three extra gates (denoted by *3 gates*) has been conducted in (20) using linear equation translation of the quiescent (static) current measurements at gate-level. Test patterns are generated via ATPG to enable the leakage

current measurement. Then the gate leakage is estimated using convex quadratic optimization with linear constraints. Full ranked matrix is solved for the unknowns which are the scaling factor. Scaling factor is the measure of the deviation of the leakage current from the nominal value using Mean Square Error (MSE). The resultant quadratic program (QP) is calibrated for the systematic variation by shifting the scale i.e. using high pass filter in MATLAB. Random variations were only left with the QP. After that, anomalies were detected by iterative algorithm using consistency metric. The algorithm re-weights the scaling factor using Gaussian kernel. After the adjustment for reweighing, the scaling factor was re-estimated using this new value. And then the iterative counter was updated. The loop terminates if the improvement in the consistency wass greater than the minimum required improvement in consistency. The consistency metric is the square distance between the initial estimate (real value) and the iterative value (estimated value) of the scaling factor. The gate with the highest differences in scaling factor was the one with anomaly. Change in scaling factor was calculated as `greatest value minus re-estimated value`. Using the differences based on maximum likelihood, one could classify new chips based on the signatures from the scaling factors of only a few gates.

Gate level characterization (timing or power measurements) is taken in account in (21). Linear equations (LaGrange polynomial) are made and processed via linear programming to find the unknowns. Solving the system of linear equations, translates the side channel characteristics of smaller gate-level structural properties. While effective, this technique does not perform well for larger chips with more gates, higher accumulated measurement noise, and more sophisticated process variation models.

In (22) Koushanfara proposed a **multi-modular non-invasive** trojan detection for gate-level hardware trojans. It locates the trojan gate and maximize its impact. Through Multi-Parameter Functional SCA, estimation of multiple measurement types as such timing, leakage and dynamic is

converted into linear equations. This helps in reweighing anomalous gates by maximizing the impact of benign gates. Reweighing is done through an iterative algorithm for each measurement that uses greedy anomaly detection. The algorithm starts while considering zero anomalies. Every anomaly is added by the greedy anomaly detection which is based on likelihood improvement. The stopping boundary of this algorithm is the improvement criteria evaluated through the greedy anomaly detection. When the trojan circuit reaches the same diminishing return difference as that of trojan-free circuit, no significant improvement is seen further. Systematic (interchip and intrachip) variation is considered as Gaussian distribution and dealt with high pass filter, leaving behind the anomalies and random variation. The number of iterations of this algorithm is much less than the number of gates. This is because not all gates were reweighed and the improvement criteria has a diminishing return property that would decrease at each iteration and makes it solvable in polynomial time. This concept of multimodal detection is combined with four different voting methods; unanimous, majority, conservative and weighed. It is tested against one extra NAND2 gate that was inserted as a trojan, and in another circuit 3-gate-comparator circuit is inserted in the IC.

The **thermal to power inversion** procedure is proposed in (23). This involves InfraRed (IR) imaging from the backside of the silicon die, and then converting the thermal image to spatial power maps. Random vectors are applied to the circuit to get the estimated power trace. Despite of the trojan type, sequential or combinational, the power consumption ratio of the trojan and the IC is the only factor that impacts the detection results. For that, packages heat spreader is removed and IR imaging techniques are applied to 5 bench circuits which gives a high resolution thermal map. Thermal features are extracted in the form of matrix from thermal maps using 2D Principal Component Analysis (2D-PCA). This is an image projection technique that makes use of the spatial correlation information to achieve better performance than conventional one-dimensional

PCA. This is applied on both authenticated chips/golden chips and the chips under test. Since the anomalies cannot be detected using inherent low pass filters of heat conduction, these features are transformed to spatial power maps using the heat equation. Using 5 different levels, 20 to 40 percent of process variation is added. This is analysed using high-order quad-tree structure. The transformed features of spatial power maps are then compared against golden chips heuristically to locate trojan. For that 3 different benchmarks, 128-bit Advanced Encryption Standard (AES) cipher, 32-bit MIPS Processor and Reed-Solomon Decoder are analysed. For each benchmark, 5 different PV levels, 10000 chips with different sizes of trojans at different locations are generated. So, with different PV levels, different trojan sizes and different Trojan locations 100,000 chips of each benchmark are generated for testing. Trojan detected through this process were 3 to 4 order less than the total power consumed by the entire IC. The comparison between the trojan-free and trojan-IC is measured by the Euclidian distance of 10 largest eigenvalues of the eigenvectors produced during 2D PCA. A kernel function is used to avoid false alarms and to set the threshold. Chips with higher power density generated more heat during the same period, which formed a larger temperature gradient. This makes the region of IC with trojan more prominent.

4.3.3 Added Test-Circuits

This class of trojan detection requires extra circuit in an ASIC design. It uses statistical analysis to verify that generated output are correct. Two of the well known mechanisms are discussed below.

Functional Testing: It is basically used to check manufacturing or unintentional faults. All the inputs of the ICs are stimulated and the output is monitored. If these values do not match to the expected values, a fault or Trojan is detected.

Usually trojans are carefully crafted that their abnormalities are not easily detected through this

test. These tests can be handy in parallel with some other detection techniques like SCA. Always-on trojans does not have much of the impact on power traces (24). In such cases rare-test-occurrences can be carried out where rather than testing all the possibilities, one test against rare events within the circuit module are considered (25).

The other reason functional test are not handy because checking for all the possibilities is not a feasible solution in most of the cases and is more likely to have brute force kind of computational complexity.

Canary Mechanism General methodology of this technique is same but it varies for different trojan types. For example, to detect abnormalities in transistor aging as such mechanism can be set to monitor the delay. The additional dedicated circuit compares the propagated delay by the transistor with some pre-stored value. The compared values are adjusted with the normal transistor age and a flag is raised when abnormality is detected. This technique does not give always perfect result as it does not take into account the aging of the test circuitry. Improvement can be made by adding these factors into calculation aswell.

Another example is comparing the checksum to detect the fault injections. This is a common method used by Intel as well. The example is studied in (1) where this mechanism is employed in Built-In-Self-Test [BIST] in Ivy Bridge microprocessors RNG. This added test-circuitry compares the resultant CRC with the pre-stored value of CRC. A flag is raised when a mismatch is found because of which that random number is discarded.

4.3.4 IC burn-in testing

Like many other reliability monitoring tests, IC burn-in aims at accelerating detection and screening out of what is known as 'infant mortalities' (early-life latent failures). It is basically a relia-

bility test which checks for the breakdown of weak parts of the chip before it is made ready for shipping. The baking process is intended to bring any change that might will occur at high voltage during the later life of IC to occur.

It is a destructive, costly and time taking process and is done with small lot of chips only to ensure the high intended quality of the complete lot of ICs.

4.4 Conclusion

At the end of this chapter it is evident that there is no ultimate way of detecting all kind of trojans. Different techniques are for different platforms and in those platforms, for refining detection, various ways are proposed to detect trojans. Even then, to get state-of art or near to that security is achieved after combining several of these methods together. This chapter discussed the research work for detection of PHTs.

Chapter 5

PREVENTIONS AND COUNTERMEASURES

5.1 Introduction

This chapter discusses prevention from counterfeit hardware and countermeasures to avoid casualties once the hardware is already subjected to PHTs.

5.2 Overview

To achieve state-of-the-art prevention against PHTs different detection techniques are combined to achieve Confidentiality, Integrity, Availability (CIA) triade. But, there is always a trade-off between the security, performance and cost. Moreover developing a countermeasure against a trojan is an arms-race. Attackers will always find a way to bypass it and new ways will be introduced to develop protection about that too (26). The aim of this chapter to mention few of the common practises and lists some countermeasures for avoiding damages caused by PHTs

5.3 Preventions

IP consumer leave it on Intellectual Property (IP) producer/vendor to create the formal model of their design, putting up reasonable trustworthiness of not adding any trojan (27). For correct implementation of consumer's needs, a formal model of design should be created for correct translation of consumer's requirements into formal-mathematical-codification in a theorem-proving language. After the IP producer writes his Hardware Description Language (HDL) through proper channel of formal-proof, it should be cross checked that whether the required properties are ful-

filled.

5.4 Countermeasures

There are many hardware and software level countermeasure techniques proposed by different authors. The main goal of countermeasure techniques should be protecting the confidentiality of data and maintaining integrity of operation of the device being protected.

Figure 5.1 sums up some counteremsaure techniques available in literature. Generally there

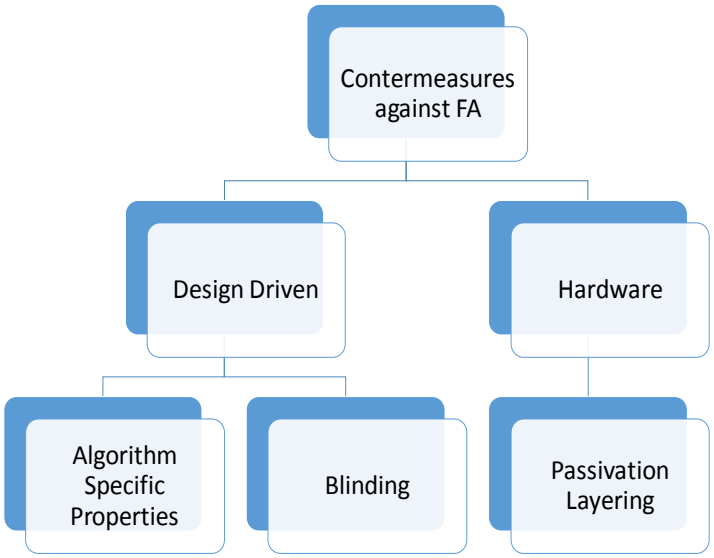


Figure 5.1: Classification Tree of Countermeasures against Hardware Trojans

exists two broad categories of how to protect against hardware trojans/ fault-injections namely hardware countermeasure and the other is design driven (6). Since the scope of this report is to find the countermeasure against post-manufacture hardware trojans, the focus will be on *design-driven* countermeasures. Considering a fault attack on AES in CTR mode, following propositions are proposed.

5.4.1 Design Driven

These techniques involve improvements at design level of implemented algorithm.

5.4.1.1 Blinding/Added Operation

Once the counter is successfully reseeded, an added layer of some random algorithm as simple as arthematic operation can introduce diffusion that will be unknown to attacker (28). It is an algorithm independent technique and has the least of the overhead.

5.4.1.2 Checking Algorithm-Specific properties

In many cryptographic algorithms especially in block ciphers there are already some means to avoid redundancy. For example the modes of operation used for implementing block ciphers. It is the requirement of the counter mode that the value of the counter should not be reused in within the encryption process. A simple added check to make sure that the current value of counter is not equal to any of the previous values can detect the similarity of the counter values. A threshold can be set to allow certain level of similarity between two counters.

5.4.2 Inherent Protection

These techniques involve improvements during manufacturing process of ICs.

5.4.2.1 Passivation layering

This is another implementation and technology independent technique for avoiding rigorous fault injection especially through Focused Ion Beam (FIB) after manufacturing process of IC. This is a metallurgical technique in which refractive protective layer is coated to create a shell against the FIB drill as it slows down the penetration of the beam by building up enough capacitive load that

fails the chip (29).

5.5 Conclusion

This chapter was dedicated to prevention and countermeasures for PHTs. Like detection techniques, one prevention or countermeasure cannot guarantee from all of the trojans but a combination of them helps to achieve the goal. This chapter not only mentioned already proposed techniques, but has proposed few of its own methods after amending the previous ones according to the need of PHT discussed in this dissertation

Chapter 6

PROPOSED MODEL FOR PARAMETRIC HARDWARE TROJAN'S DETECTION

6.1 Introduction

In this chapter proposed model for detection of PHT, discussed in our case study is explained in detail. It takes the concepts of previous chapter to next level. Two models are proposed based on *Pearson Correlation* and *Column-wise subtraction*. These two concepts are also discussed is this chapter.

6.2 Analysis of Detection Techniques

Four ways to detect hardware trojans were discussed in section 4.3. Each had its own merits and demerits. Choosing from the detection technique requires understanding the behaviour of the target that is to be detected. It also depends on the available resources and required security.
As discussed in section 7.2, we have visualised the trojan at HDL level. For this reason detecting trojans through 'optical inspection' is an invalid preposition. The reason is clear, PVC detects on-chip trojans. Also, PVC is very expensive technique that requires exquisite equipment, as discussed in section 4.3.1, and the equipment was not an available. However, PVC is the most relying detection technique for detecting stealth PHTs. Second technique to detect hardware trojans is 'functional testing'. This technique also includes other error propagation checks that are deployed to avoid error propagation, just like CRC does. This technique can also detect

dopant trojans but may require exhaustive test pattern space. Change in output can be mapped from expected result which will indicate the presence of trojan. But the reason for not choosing this technique is required computational complexity which makes detection equivalent to brute force. Another reason is, ICs that are compromised with dopant trojans, are usually designed in such a way that the trojan also compromises these functional tests. ThThis can be done by carefully crafting faulty CRCs.

For this reason we have directed our scope towards Side Channel Analysis (SCA). This is a detection technique that can be used to analyse different devices with small changes in the general procedure. SCA can also be used as a last-mile-solution, as trojans always leave their footprints on side channels. However in nanotechnology these footprints merge so well with the intrinsic behaviour that it gets difficult to find them. Advance power analysis techniques are then used to solve this challenge.

6.3 Proposed SCA Model for detecting PHT

Based on the analysis on different detection techniques, an SCA model is proposed for detecting the dopant level PHT.

Although this method is mostly used for attacking purpose but we can use same statistical technique for detection. We have selected consumed power as the side channel for our analysis. Formal steps of conducting typical SCA are same as in 4.3.2.2. In this section we will only discuss the improved steps in detail.

6.3.1 Choosing an intermediate state

: The intermediate state that we have selected is the first output byte of S-box in the first round of AES-128. This intermediate result is the function of first byte of counter and first byte of the

round key. The function can be written as AES **(c,k)** where **c** is the counter and **k** is the key.

There are two reasons for choosing output of the S-box. First, because of iterative nature of the encryption process, same registers of S-box are used to store data in 1st, 2nd, 3rd till 10th round. Use of same register multiple times give rises to ghost peaks that occurs due to correlation with these shifted values of s-boxes. The relation of ghost peaks and s-boxes is discussed in detail in (30).

Usually, when we have access to the plaintext, we start our cryptanalysis process from first round to expedite our attack. In case its cipher text, we attack the last round as it makes it easy because of same reason. Although in our case we have every knowledge about inputs and outputs of the cipher, but for detecting trojan in counter, we will choose 1st round as minimum effort will then be required to recover the actual counter value.

6.3.2 Taking Power Traces

In second step, power traces are acquired. For this, power consumed by the device for encrypting or decrypting data blocks is measured. In typical SCA, for each acquired power trace, corresponding data values of selected intermediate function are known 4.3.2.2.

In this case, s mentioned in 3.4, counter values are not known because of inserted trojan, but the key is known. In our AES code, the counter value is changed with every AES encryption. These counter values are written as vector $c_i = (c_1, c_2, c_3.c_D)$, where c_i denotes counter value in the i^{th} encryption or decryption.

During each encryption or decryption, power trace are recorded. The power trace corresponding to c_i is denoted as $t_i = (t_i, 1.....t_(i, T))$ where 'T' is the length of the trace. Thus while recording each power trace against each counter value, a matrix of size **C x T**, where **C** is the data set of counters, will be constructed. All these traces are aligned.

This process is carried out with both AES codes discussed in 7.2 and two resultant matrices are formed, matrix **A**, that is formed against trojan-free AES code and is of size **CxT** and a matrix **T**, that is formed against trojan-infected AES, *AES-128-trojan* code and is of size **D x T** where D is the data set of trojan-counter with d_i as the trojan-counter value at i^{th} encryption or decryption.

6.3.3 Comparing Power Traces

In last step of SCA, power trace are compared against various attributes of the algorithm, such as different algorithmic operations, to detect desired differences between power traces. In this case the desired result of comparison is to show us bytes of the counter that are fixed /manipulated. And hence it will be proved that the counter value is different.

6.3.3.1 Pearson Correlation:

Pearson correlation is a statistical term that is used to measure the dependence of two data sets. This dependence is known as *Correlation*. Peasron correlation measures the *linear* correlation and its values ranges form -1 to +1, where 1 is total positive correlation, 0 is no correlation and -1 is negative correlation as shown in figure 6.1.

Full name for Pearson Correlation is the *Pearson Product Moment Correlation or PPMC*. The mathematical expression for this is given in equation 6.1.

$$r_{i,j} = \frac{\sum_{d=1}^{T}((a_{d,i} - \bar{a_i}).(t_{d,j} - \bar{t_j})}{\sqrt{\sum_{d=1}^{T}((a_{d,i} - \bar{a_i})^2.(t_{d,j} - \bar{t_j})^2}} \tag{6.1}$$

where

$\bar{a_i} = \frac{i}{n}\sum_{i=1}^{n} a_i$ and so is t_j, the mean.

$r_{i,j} =$ element in i^{th} row and j^{th} column of Resultant matrix **R**.

$a_i =$ element in column **i** of matrix **A**.

$t_j =$ element of matrix **j** of matrix **T**.

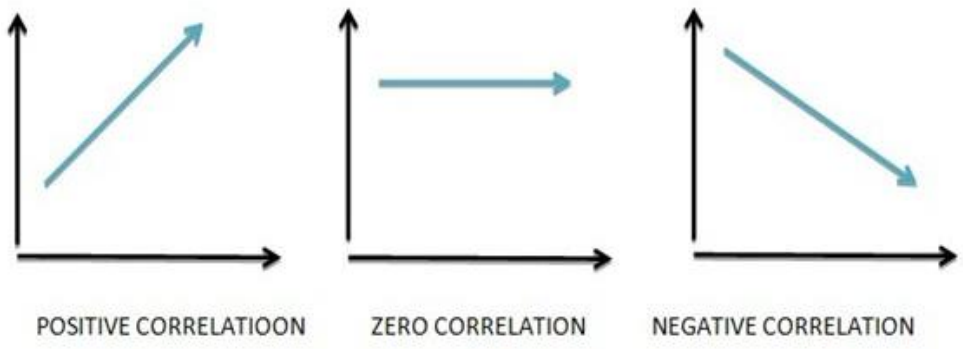


Figure 6.1: Pearson Correlation, Photo Courtesy: MBASKOOL

$T =$ length of trace.

The description of these matrices and its elements are further explained in section **??** which expalins the architecture of implemented trojan and in the following section.

SCA model based on Pearson Correlation

To spot the difference introduced by trojan, each column a_j of matrix **T** is compared with each column t_j of matrix **T**. This means that we will compare the power consumption of trojan-free AES with trojan-infected AES at every position. The result of this comparison is matrix **R** of size **T x T** trojan. This comparison is based on the correlation coefficient. Pearson correlation is used which is mathematically defined in equation 6.1.

In matrix **R**, higher values of $r_{i,j}$ represents better columns matching of a_i and t_j. Ideally both matrices should match, resulting in high correlation. This is because both matrices contain power traces of same counter value encryption. But since the first byte of AES-128-trojan is manipulated in our case, this change will appear in simulations as well. As a result these two matrices will

not correlate perfectly. Traces representing first byte of the counter will differ while rest of them will correlate. Therefore initial values of matrix **R** will be low, representing weak correlation while rest of the values will be high showing high correlation. We can then reveal the *index* of the manipulated bytes and the moment of time by looking at the lowest value in the matrix **R**. The indices of this value are the result of this analysis. That MATLAB code is given in appendix A.

6.3.3.2 Column-wise subtraction:

The difference between two signals can be used as a tool to detect trojans. The section below how we can use this concept. Although it is very simple way but it can be refined for different complex scenarios.

SCA Model Based on Subtraction

If two signals are same, the result of their subtraction should be zero or should have very small values. However if two signals differ, their difference is shown by some value in the remainder of subtraction.

There are two ways to use this concept of difference for purpose of detecting trojan-signal. One way is to subtract two trojan-signals, **t1[t]** and **t2[t]** that are encrypted against two different counter values **c1** and **c2** respectively. The resultant difference signal **r[t]** between these two signals should have huge peaks representing difference between two values. But instead, small values will appear where the counter byte was fixed which will represent a minimal difference between two signal values inferring similarity between two signals.

Mathematically, corresponding matrices **T1** and **T2** against **t1(t)** and **t2(t)** are column wise subtracted to get resultant matrix **R**. The columns against fixed byte will have small values representing similarity in counter value whereas rest of the columns will have large values representing difference between counter values.

Second method is to subtract a trojan power signal **t1(t)** from an ideal power signal **t2(t)**, both of which are result of encryption against same counter value, **c**. The ideal expected resultant signal **r(t)** should be a difference-signal with small to no peaks representing small or no difference ib counter values representing the counter values are same. But instead, trojan-signal **t1(t)** vary because of fixed bits. There will be huge peaks representing large difference values. This shows that the two signals, that were supposed to be same, are not same and an anomaly exist. In matrix representation, **T1** and **T2** are column wise subtracted to get **R**. The columns against fixed bits will contain large values representing similarity between two matrices whereas rest of the columns will have small values representing similarity of counter values.

Second model proposes a clever use of simpler way of detecting power traces that vary from the expected result.

6.4 Conclusion

In this chapter we have proposed SCA based model for detection of dopant level PHTs discussed in Intel's case study in previous chapters. Two models were proposed. One was based on Pearson correlation, a common concept used in statistics to find correlation between two data sets. The other model was a clever use of simple subtraction between two signals. Although in more complex scenarios the subtraction model is not very effective yet it gives the basis of concept which can be further refined.

Chapter 7

EXPERIMENTAL RESULTS OF PROPOSED SIDE CHANNEL ANALYSIS MODEL

7.1 Introduction

This chapter analyses the effectiveness of proposed SCA model for dopant PHT on Intel's Ivy Bridge microprocessor RNG through experiments. First section discusses the architecture of trojan based on case study. It is to be noted that the scope of this research is not trojan insertion but trojan detection. For this reason, hardware trojan is visualized through changes in HDL code and SCA is used to detect trojans in real time. Next section discusses real time implementation of trojan and its detection. Due to various limitations mentioned in section ??, real time power traces could not be acquired and instead power traces form online repository mentioned in ?? were taken. these traces were trianed over the proposed MATLAB model and analysis of obtained results is described in this chapter aswell.

7.2 Architecture of Trojan

As mentioned, hardware trojan discussed in (1) is visualized as code modification. The effect of trojan is that the first byte of counter in AES-128 in CTR mode is fixed by manipulating transistors of the register containing 128-bit counter value and cipher key. This trojan efficiently reduces the computational complexity form actual complexity which is $1/2^{128}$. More details f AES-128 and its counter mode can be read in section 3.3.1 and section 3.3.2.

Since, in this case, the key is either known, public or the register storing the key is manipulated to remain constant. The complexity of encryption in counter-mode depends on 128-bit secret key and 128-bit unknown counter but by taking the assumptions, now the complete complexity is dependent on 128-bit counter. The attacker will design a trojan in such a way that the complexity is reduced to n-bits. He will do this by fixing **s** flip-flops of the register containing the counter value and leaving rest of $128 - s = n$ flip flops unchanged. Each flip-flop stores one bit. This means now the complexity is reduced to **n** bits only. On the other hand, for an evaluator who does not know about trojan-constants, the counter value looks random and legitimate. This is because AES generates outputs with very good random properties, even if the inputs only differ in few bits . AES-128 pipeline code in Verilog is taken from Opencores (31) and implemented in CTR mode. Usually the counter value is incremented by 1 after every encryption and the counter register is reseeded after the number of encryptions set by the implementer. To speed up our detection process and to jump to the point-of-interest, we made our counter to reseed itself after every encryption. This means that every plaintext will be encrypted with a complete new 128-bit different counter value. This version of AES is named here as **AES-128**. We will refer to the manipulated code as **AES-128-trojan**. This code is similar to AES-128 but the first byte of counter is fixed/manipulated to 128 decimal value while the remaining bytes are left as it is to follow the routine code.

Both codes are implemented on Xilinx FPGA development board with 50 MHz clock cycle.

7.3 Experiment Setup

This section discusses steps that were carried out to perform SCA experiment. Two FPGA development boards, Spartan 3E and Virtex 5 were used and analysed for this purpose. During the analysis, few improvements were required to acquire power traces. It was observed that these

development kits are not suitable for carrying out SCA. Also, even after few improvements there are still some limitation which hindered successful SCA. All these observations and limitations are discussed in this section.

7.3.1 HDL code

As a first step, verilog code of AES 128 is simulated in Xilinx ISE 12.3. Top module contains three other modules as shown in figure 7.1 and explained below

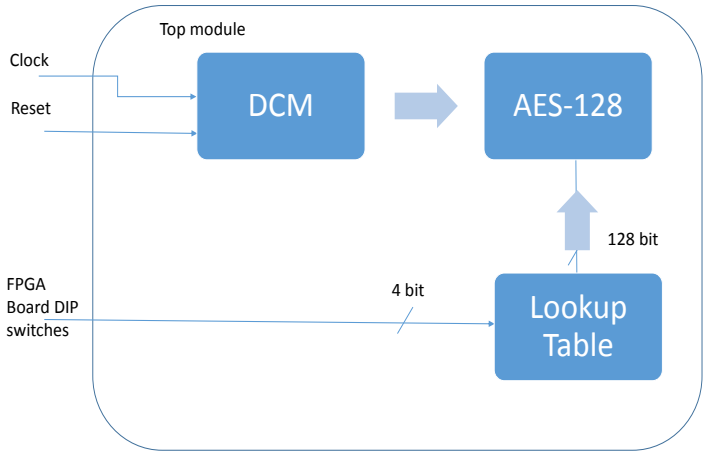


Figure 7.1: HDL Code Block Diagram

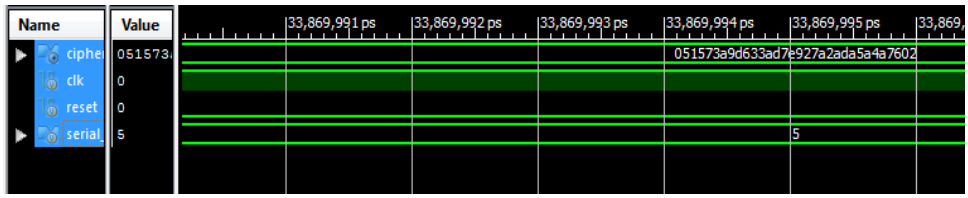
1. Digital Clock Manager (DCM): Clock divider module bring down 50 MHz on board clock to 5 MHz.

2. Lookup Table: This module selects from 16 counter values from the lookup table through 16 possible combination of the DIP switches on board. The reason for choosing only 16 values is because of only 4 DIP switches available on board that could give only 24 =16 (0

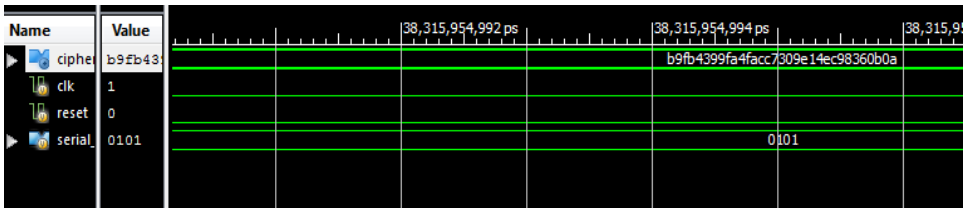
- 15 decimal) options.

3. AES-128: Third module is the main module i.e. AES-128. Upon reset, default value of counter that is zero is encrypted at 5MHz.

To make it sure that the verified code is NIST verified, Known Answer Test (KAT) was also carried out. Xilinx Simulation is shown in figure. It can be seen in figure 7.2 that how the ciphertext changed when same counter value is encrypted trough trojan-AES and trojan-free-AES code. The plaintext value at counter value 5 (0101 in binary) is 'hffffffffe000000000000000000000000'. The ciphertext value is different in case of both AES-128 codes representing trojan.



(a) AES-128 with no trojan



(b) AES-128 with trojan

Figure 7.2: HDL Simulation of AES-128

7.3.2 Test Bench

After coding, selection of right FPGA is very important. There is a preference trade-off over here. If one is peculiar about FPGA board, then he should code the algorithm accordingly or, if the code is not to be changed, like in this case, then the board should be selected accordingly.

The selection of board is dependent on size of an FPGA that can accommodate the algorithm size. Other selection merits include but not limited to are gate density, performance, cores and memory. Figure 7.3 shows the hierarchy of FPGAs.

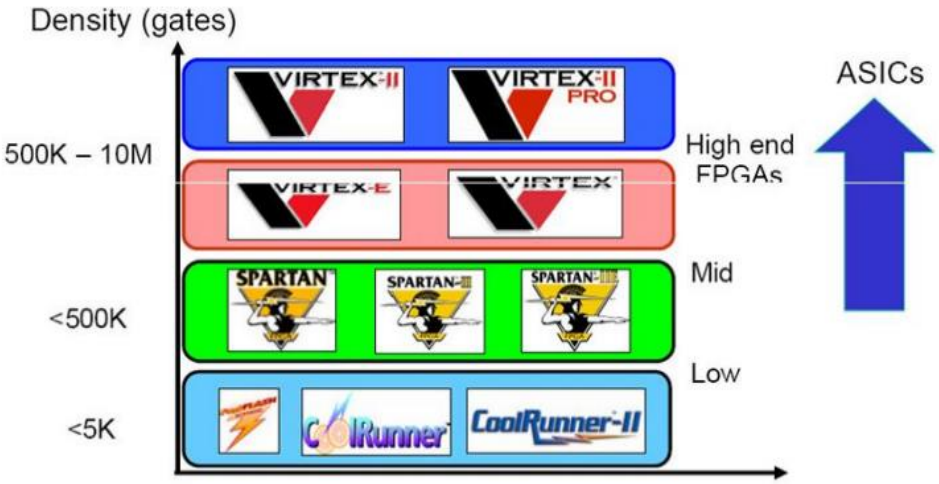


Figure 7.3: FPGA Types

There are few consideration that are to be made while deciding to acquire power traces. These considerations are discussed below:

1. Choosing an appropriate site for measuring dynamic current:

   On board power supply is usually divided into multiple smaller power rails usually 3 to 4. These power rails supply voltage to different components of board such as external component, LCD and other auxiliary component and to the core logic of FPGA. The main interest for power analysis is on the power rail of core logic. It is therefore important to identify that IC and the jumpers across it.

2. Removing decoupling capacitor:

   The purpose of decoupling capacitor is to remove noise signals. While performing power analysis, we are considering every aspect of power consumed and therefore d do not want any component to remove or attenuate any signal component on its own.

3. Choosing appropriate values of external components (resistor and capacitor) to measure voltage in case of voltage measurement.

   In case passive voltage probes are used for measuring voltage drop, few changes are required on board. For measuring voltage drop, a resistor has to be placed between the global supply of board and the supply pin. The values of this resistor should be carefully selected. Although higher value of this resistor can give higher voltage swing and good voltage trace but this will give less voltage to rest of the circuit effecting its performance. Therefore a resistor of sufficient value should be used which gives appropriate voltage swing across it and also not effect rest of the circuit aswell. To avoid voltage overshoot at clock transitions, a capacitor is to be connected in parallel with the resistor. Values of both capacitor and resistor are measures through hit and trail method.

7.3.2.1 SASEBO

SASEBO stands for Side-channel Attack Standard Evaluation BOard. This board is dedicated for research in side channel analysis. Unlike in development kits, SASEBO has multiple options to perform successful SCA with much ease. for more details about SASEBO boards refer (32).

7.3.3 Oscilloscope

Once the HDL code is burned on FPGA, AES encryption is triggered through DIP switches on board. For acquiring power traces, power consumed is measured by the probes of oscilloscope

which is them sampled and stored in memory. Key features to be noticed while choosing an appropriate oscilloscope includes sampling rate, deep memory, bandwidth and triggering mechanism.

10+1 rounds can be seen in the sample power trace of AES-128 figure 7.4. This figure shows a

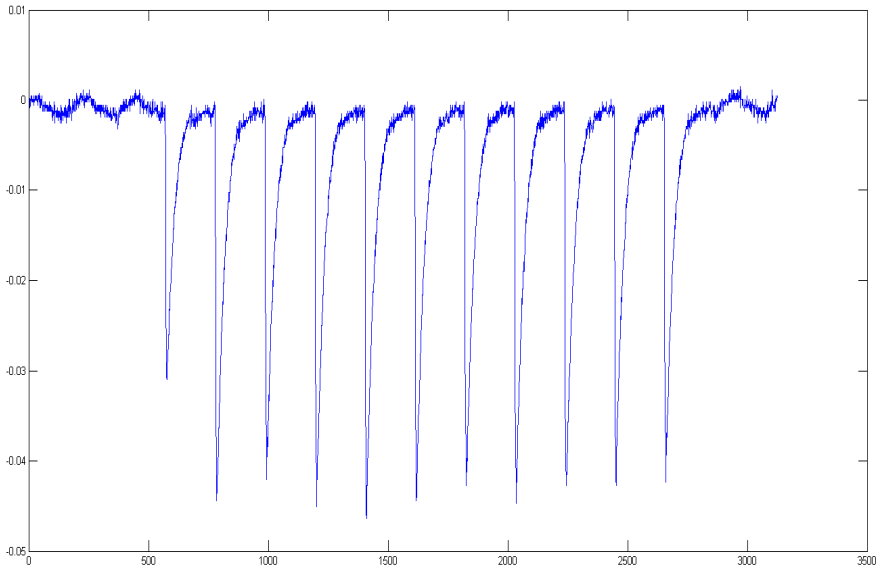


Figure 7.4: Power trace of AES-128 encryption, (Photo Courtesy: DPA Contest)

sample power trace against single counter value that was encrypted. The dips here are showing the excessive power consumed by FPGA during the four stages in each round of AES-128.

7.3.4 Probes

The probes connected with the power supply of FPGA and oscilloscope monitors and records the sampled power traces of power consumption during the AES encryption at runtime. The sensitivity of probes is very important. Also it is important to know what kind of probes are suitable to the experimental setup. Brief knowledge of different kind of proves is discussed below. There are basically two type of oscilloscope probes available, active probes and passive probes. Although active probes are very expensive but they are recommended over passive probes because

of following reasons:

7.3.4.1 Active probes

1. High impedance ($<$ 2pF typical), high bandwidth.
2. Prevents unwanted loading on signal.
3. Requires short GND wire for best performance.
4. Used for AC parameters, probing devices In-system.
5. Usually 10:1 signal attenuation.

7.3.4.2 Passive probes

1. More impedance than active (6 to 9 pF typical).
2. Causes additional loading.
3. Used for measuring timing parameters, frequency, measurements not affected by loading .
4. Usually 10:1 signal attenuation.

Acquired traces are then processed through MATLAB for detecting trojan. For this purpose, the oscilloscope was connected with the PC which is shown in general block diagram of experimental setup in Figure 7.5.

7.3.5 MATLAB Simulation

The communication between oscilloscope and FPGA board is done through a laptop. All the statistical analysis is carried on MATLAB running on this laptop. The MATLAB code will show

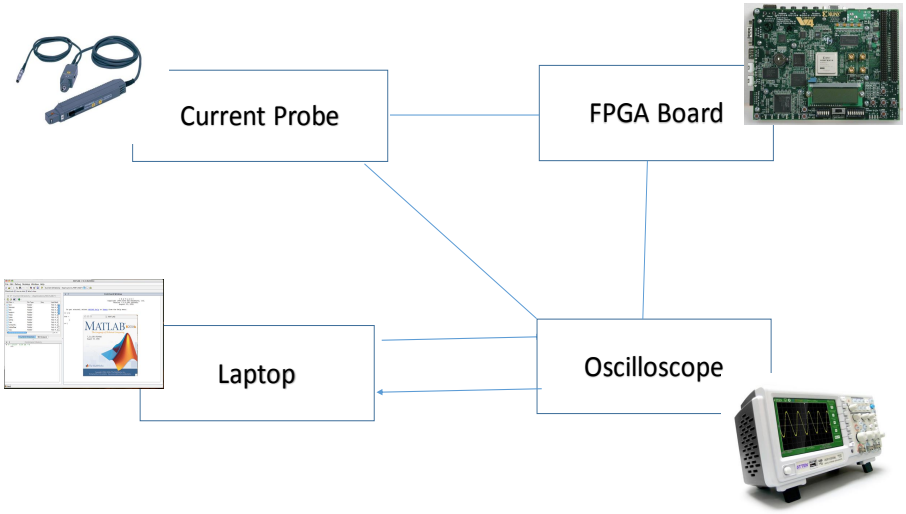


Figure 7.5: Basic Block Diagram of Experimental Setup

relation of two versions of AES code. Genral procedure of the code is shown in figure 7.6. When the code is provoked, it will process two data sets in matrix form, named matrix A and matrix T. One of the chosen proposed SCA model will process these data sets and gives resultant matrix R as its output. This matrix is then plotted to identify high and low peaks easily.

In case of correlation model and subtraction model, low peaks shows that there is less correlation or small difference respectively between two data sets of traces. And high peaks shows stronger correlation or higher difference respectively. The low peaks are due to the reason of change of counter values which, in ideal case, should be high. Higher correlation shows that two data sets are against the same counter values and therefore the resultant intermediate correlate.

It is to be noted that when two similar power traces are subtracted, there should be no peak visible. These small peaks shows the remainder of the subtraction operation. Higher peaks shows higher differences between the two traces indicating intentional/trojan values of the counter.

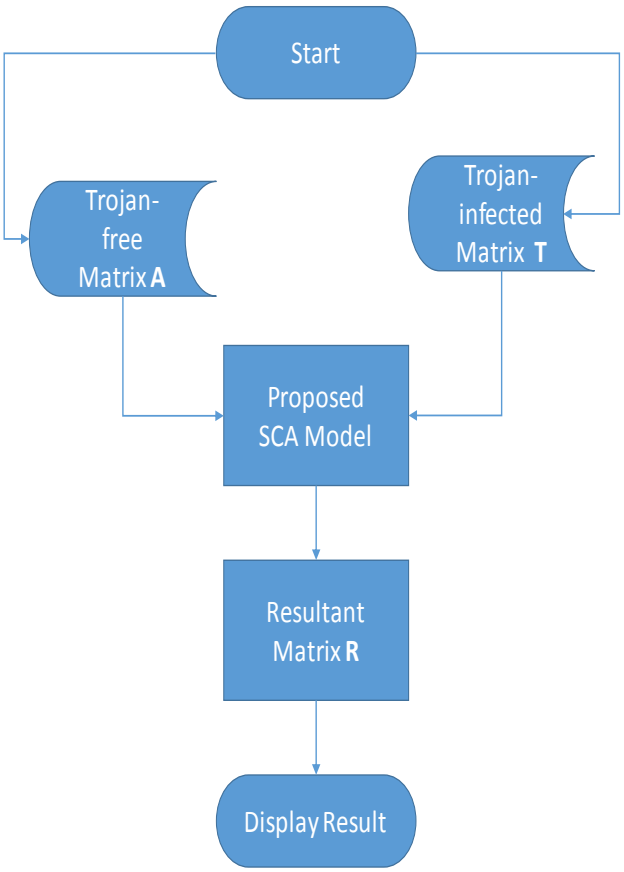


Figure 7.6: FlowChart of MATLAB code

7.4 Limitations

This section lists all the limitation observed that made acquiring power traces difficult.

1. The HDL code could not be run over an FPGA development kit because of various limitations. First, Virtex 5 development kit did not have jumpers across its input voltage IC from where voltage or current drop could be measured.

2. Passive probes are not suitable for measuring current in micro amperes.

3. Because of unavailability of jumpers, external component like resistor could not be con-

nected across which voltage drop could be measured.

In order to analyse the effectiveness of SCA for detecting parametric trojans, offline traces of consumed power during AES-128 encryption were considered. These traces were taken form online repository as mentioned in introduction of the chapter.

7.5 Results Analysis

To explain the results of proposed trojan detection scheme, the MATLAB code was trained over these offline power traces. Figure 7.7 and figure 7.8, at the end of chapter, shows the result obtained after running through MATLAB code.

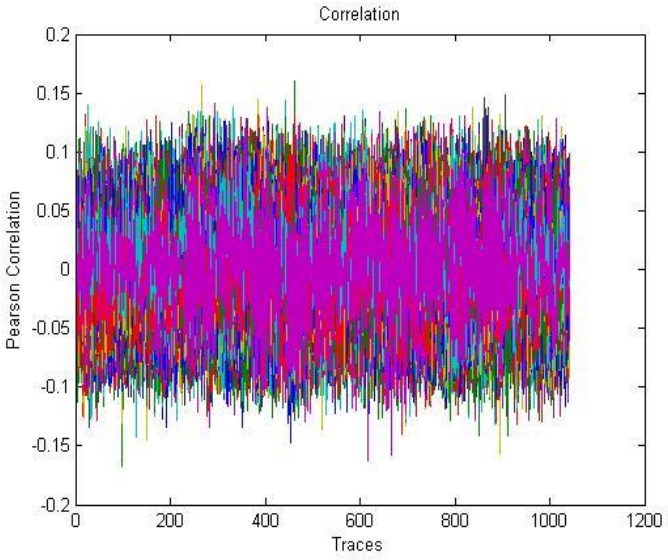
Figure 7.7 shows correlation between different counter values after AES-128 encryption. As can be seen in figure (7.7a) all peaks are uniformly distributed. High peaks show high correlation between two or more traces. The sampling rate was 1024 samples per second. and 1024 different counter values were considered.

In second case there was a linear relation introduced in second column of same power traces. This was done as to visualise effect of parametric trojan. Also as discussed earlier, parametric trojans were inserted by fixing the gates which adds consistency and effects the output in a constant manner. That constant effect is visualized here by introducing a linear function in column two of power trace. The result can be visualized in figure (7.7b). There is a clear constant high peak through out the output of correlation. this constant high peak represents there is high correlation between the traces against different counter values. And this correlation exists between all traces. In practical scenario, same constant peak is found but is of small amplitude and therefore is hard to distinguish. Different refining techniques discussed in Chapter 3 can be used to further refine results and to clearly identify this constant peak.

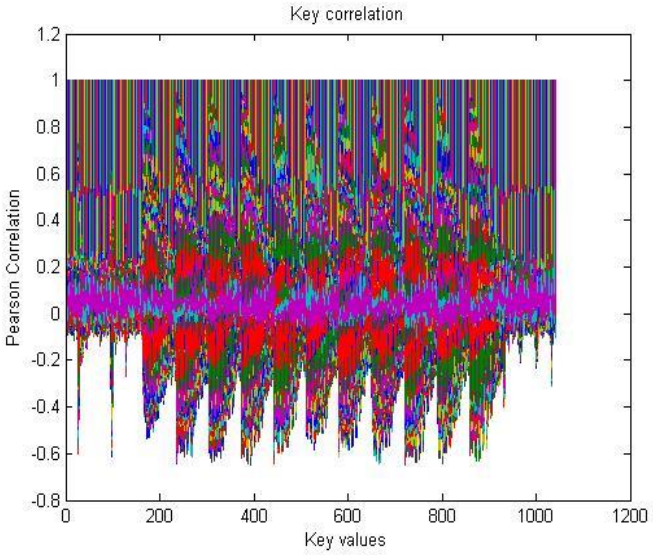
Figure 7.8 shows the same concept but in this figure rather then multiple traces, only two counter values and their respective power traces were correlated and result is obtained. Figure (7.8 a) is correlation between two power traces of AES-128 encryption. High peaks and low peaks are uniformly distributed. but in figure (7.8 b) constant peaks can be seen showing consistency between two power traces.

7.6 Conclusion

This chapter has listed down the observations and limitations of detecting dopant trojan. Also through results we have analysed the effectiveness of SCA in detecting such trojans. The effect of trojan in power traces was explained with MATLAB simulation.

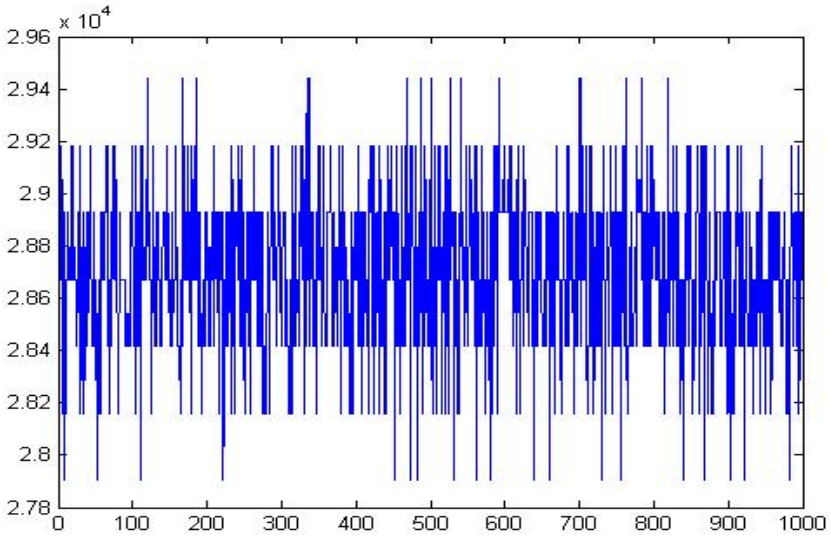


(a) AES-128 with no-trojan

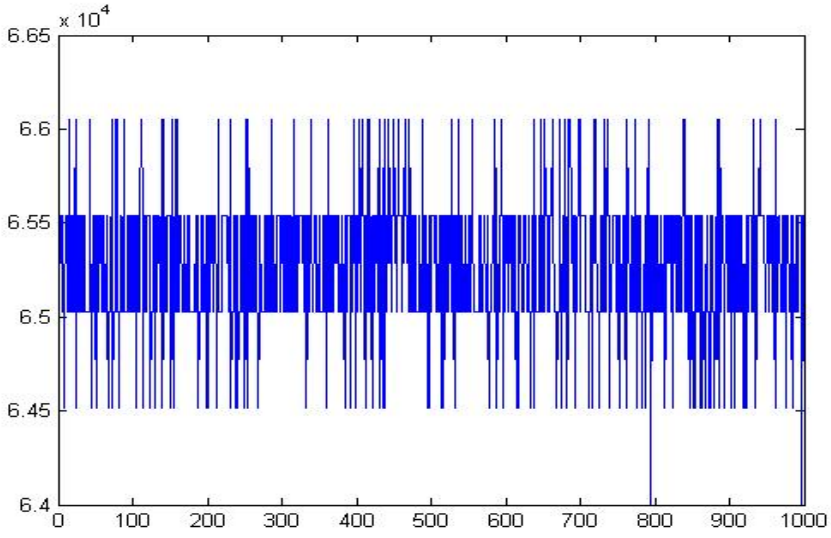


(b) AES-128 with trojan

Figure 7.7: Pearson Correlation of Multiple Traces



(a) AES-128 with no-trojan



(b) AES-128 with trojan

Figure 7.8: Pearson Correlation of Single trace

Chapter 8

CONCLUSION

8.1 Introduction

This is the last chapter of thesis report. It aims to briefly describe how the objective of research were concluded and gives suggestions for future work.

8.2 Conclusion of Research

This research tightly follows its five main objectives, giving an in-depth knowledge of Parametric Hardware Trojans. A detailed and comprehensive survey of hardware trojans, their detection and prevention was carried out. A *taxonomy* was proposed based on this study. After going through the literature review of *hardware trojans*,different detection techniques for PHTs were analyzed for PHTs. Although the report discusses different detection techniques for different kind of PHTs, *SCA Model* was proposed for detection of dopant level PHTs.

The study was not only limited to theory but all *experimental results* were also carried out. After visualization of hardware trojan in verilog, different FPGA boards were tested for side channel analysis. It was observed that common FPGA development kits were not suitable for SCA and all the observation and limitation of these boards were discussed. At the end Side Channel Attack Standard Evaluation Board (SASEBO) were also suggested to carry SCA to next level. Alot of work on them can already be found in international research institutions.

8.3 Future Work

SCA is a very important domain of cryptography. Its an important aspect of hardware security and a pivot point of research in cryptography. Alot of research is already carried out and various vulnerabilities have been discovered. More focus in other types of hardware trojans is required, giving an awareness about other hardware security aspects. Not only its defensive side but offensive side should also be studied as it gives an insight of hardware trojans. therefore focus on insertion techniques of trojans is also an important domain. Studying SCA in detail on SASEBO is an important and interesting topic of research,