HARDWARE TROJAN DETECTION IN THIRD-PARTY DIGITAL IP CORES

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY

IN

VLSI DESIGN AND EMBEDDED SYSTEM

BY

RAKESH CHANAMALA

213EC2195

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

ODISHA, INDIA. 769008

2015

HARDWARE TROJAN DETECTION IN THIRD-PARTY DIGITAL IP CORES

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY

IN

VLSI DESIGN AND EMBEDDED SYSTEM

BY

RAKESH CHANAMALA

213EC2195

Under the Guidance of

PROF. KAMALA KANTA MAHAPATRA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

ODISHA, INDIA. 769008

2013 – 15

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ODISHA, INDIA- 769008

CERTIFICATE

This is to certify that the thesis report entitled

HARDWARE TROJAN DETECTION IN THIRD-PARTY DIGITAL IP CORES

submitted by

Mr. Rakesh Chanamala

bearing roll no. **213EC2195**
in partial fulfilment of the requirements for the award of

MASTER OF TECHNOLOGY
in
VLSI DESIGN AND EMBEDDED SYSTEM

during session 2013-15 at National Institute of Technology, Rourkela, is an authentic work carried out by him under my supervision and guidance.
To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other university/institute for the award of any degree or diploma.

Prof. Kamala Kanta Mahapatra
Department of ECE,
National Institute of Technology,
Rourkela.

Date: May 30th 2015

Dedicated to
National Institute of Technology, Rourkela

CONTENTS

ABSTRACT: v

List of Figures vii

List of Tables viii

LIST OF SYMBOLS AND ABBREVIATIONS ix

1. OVERVIEW 1

1.1. INTRODUCTION 2

1.2. MOTIVATION: 7

1.3. RESEARCH OBJECTIVE: 8

ORGANIZATION OF THE THESIS: 9

2. VERIFICATION METRICS FOR TROJAN DETECTION 10

2.1. SYSTEM ON CHIP VERIFICATION FOR TROJAN DETECTION: 11

2.2. VERIFICATION v/s TEST: 11

2.3. VERIFICATION METRICS: 12

2.3.1. FORMAL VERIFICATION: 12

2.3.2. COVERAGE METRICS: 13

2.4. HARDWARE TROJAN DETECTION PROBLEMS AND SOLUTIONS: 16

3. THE AES AND RSA ALGORITHMS 23

3.1. ADVANCED ENCRYPTION STANDARD (AES): 24

3.2. RSA: 31

4. TROJAN BENCHMARKS AND THEIR STRENGTHS AND WEAKNESSES 34

4.1. AES TROJAN BENCHMARKS: 35

4.2. POWER AND AREA ANALYSIS: 39

4.3. STRENGTHS AND WEAKNESSES OF AES BENCHMARKS: 42

4.4. RSA TROJAN BENCHMARKS: 47

4.5. STRENGTHS AND WEAKNESSES OF RSA BENCHMARKS.....48
4.6. Trojan Detection in RSA-T300: 50
5. DESIGN OF NOVEL AES TROJAN BENCHMARKS.....54
5.1. EXISTING DIFFICULTY:.....55
5.2. PROPOSED NOVEL BENCHMARK DESIGN: 58
6. CONCLUSION AND SCOPE OF FUTURE WORK:.....60
6.1. CONCLUSION:.....61
6.2. FUTURE WORK:.....62

ACKNOWLEDGEMENTS

I have procrastinated (Procrastinate: verb. postpone the action, The Oxford Dictionary) writing this passage to the most recent moments prior the printing of this thesis, because I assume, it is the toughest and undeniably the most read part of this dissertation. According to the synaptic information stored in my Central Nervous System (CNS), it has been forty-one days since I composed the first sentence for this manuscript, nearly thirteen months (precisely talking, 392 days) since I started the research which steered to this and altogether almost twenty-two months since I officially got associated with NIT Rourkela for my post-graduation studies. And what this entire period of time have been; packed with profound knowledge and splendid experience, hard work for fun, enjoyment and frustration, teamwork and friendship. At the end of this this wonderful period of time, I wish to give away a vote of thank to all the people who held my hand for pursuing an M. Tech degree. First and foremost, I am grateful to my research advisor Prof. K. K. Mahapatra, for the opportunities he spawned for me, for promoting to learn the topic absolutely new for me, for his invaluable advice on the big or small problems, for his encouragements, and for believing my abilities throughout the period. I am also greatly indebted to Mr. Sudeendra Kumar, a Ph.D. scholar at NITRKL, for introducing the exciting topics in this domain to me, an infant M. Tech student who started journey on unknown paths in this space. Without his knowledge and priceless guidance for this research, boundless efforts and patience, and long lasting technical discussions, this research would have never been fruitful. Many thanks to both of you for such unmatched support. I express my whole-hearted gratitude to Prof. D. P. Acharya, Prof. P. K. Tiwari, Prof. A. K. Swain, and Prof. N. Islam for their thoughtful teaching and suggestions during my courses in M. Tech and making available all necessary facilities and infrastructure for iii studies. I am also thankful to all research scholars in VLSI lab and all other labs for maintaining the lab, availing access to the same 24 ×7 and creating vibrant atmosphere for research. Being an M. Tech student here, I had a privilege of

getting and working together with many of my peers nationally. Stay with such diversified classmates has made it more delighted and memorable. Thanks to Anil Kumar, one who always give a try, Hanumantha Rao , a mythologist, Siddhardha, a creative mind, Krishna Reddy, a stubborn personality and hard worker, Sailaja, an intelligent girl and Sri Kanya, an innocent girl for making joyful learning here. Finally, I owe my heartiest gratitude to my father Prabhakar Chanamala, mother Mrs. Swarajya Lakshmi Prabhakar and sister Miss. Divya Sruthi Prabhakar for their unconditional support, love, inspiration and sacrifices.

Rakesh Chanamala

ABSTRACT:

Due to Globalization outsourcing of SoC designs either for verification, testing and fabrication has become inevitable. Modern System on chip (SoC) is complex process. Modern SoC's can be designed time effectively and cost effectively with the help of third party Intellectual Property (IP) core vendors. Various processors cores (like ARM, Power PC), communication controllers (CAN, Zigbee) and control cores (PWM, Analog comparator) will get incorporated into SoC's, which are supplied by different vendors. The original SoC manufacturers are IP integrators, targeting a particular application.

In this process, various issues like IP protection, IP rights and problem of malicious IP's will arise. Recent addition in this list is Hardware Trojans (HT). HT's can be included by rogue designer in design house or at overseas fabrication factories. The objective of these HT's includes manipulating the functionality of the chip, leaking confidential information and destroying the system. HT's included in the design phase must be weeded out during verification phase. Still now, there is no concrete method or golden rule in the existing verification framework to detect the HT's. Various verification metrics like code coverage, functional coverage and verification methodologies like OVM or UVM will be helpful in detecting HT's. Formal verification is also useful. A comprehensive framework using all verification metrics is very much required to detect HT's. We will address this issue in our thesis.

Secondly, static timing analysis (STA) and power analysis (PA) can be used to detect HT's included at both design phase and also in fabrication. In our proposed framework, we will incorporate verification metrics, formal verification, STA and PA to detect HT's.

In this report, we apply DFT techniques and standard verification metrics to detect the hardware Trojans. The microprocessors and cryptographic designs are most vulnerable for

hardware Trojan attacks. The Advanced Encryption Standard (AES) and RSA Trojan benchmarks from Trust Hub are used to verify the existing test principles like stuck at fault (SAF), path delay faults (PDF) are capable of detecting Trojans in Benchmarks. Results and analysis is presented in this report.

Also Novel Trojan Benchmarks designs were proposed to eliminate the existing weaknesses in AES Benchmarks.

List of Figures

Figure 1-1 Inclusion of 3PIP's (malicious) in SoC 3
Figure 1-2 Trojan Structure 4
Figure 1-3 Detailed Trojan Taxonomy [2] 5
Figure 2-1 Model Check block diagram 12
Figure 2-2 Equivalence Check block diagram 13
Figure 3-1 AES Encryption Block Diagram 24
Figure 3-2 AES Decryption Block Diagram 24
Figure 3-3 AES encryption for 128-bit plain text 25
Figure 3-4 various steps in one Round of AES[21] 26
Figure 3-5 State vector matrix before and after shift rows operation 27
Figure 3-6 Mix Column layer 28
Figure 3-7 Key Expansion 29
Figure 3-8 The Key Expansion Flow [21] 30
Figure 4-1 Trojan inserted AES [lin] 36
Figure 4-2 CDMA based TSC implementation [lin] 37
Figure 4-3 TSC using unused pin for RF transmission 38
Figure 4-4 Coverage report for RSA-T200 52
Figure 5-1 Simulation waveforms of AES-T500 for un-triggered Trojan 56
Figure 5-2 Simulation waveforms of AES-T500 for Triggered Trojan 56
Figure 5-3 The structure of Trojan in AES-T500 benchmark 57

List of Tables

Table 2-1 Verification versus test.....11
Table 4-1 AES Benchmarks Description 35
Table 4-2 Dynamic Power and Area of AES Benchmarks 40
Table 4-3 Summary of Detection of AES Trojans 46
Table 4-4 RSA Trojan benchmarks description 47
Table 4-5 RSA Trojan benchmarks overall detection summary 53
Table 6-1 Trojan Category and its strength.....61

LIST OF SYMBOLS AND ABBREVIATIONS

The following is the list of abbreviations that are encountered in this thesis.

AES	Advanced Encryption Standard
ASIC	Application Specific Integrated Circuits
ATE	Automatic Test Equipment
ATP	Automatic Test Patterns
ATPG	Automatic Test Pattern Generation
CMOS	Complementary Metal Oxide Semiconductor
DC™	Design Compiler
DFT	Design for Testability
DUT	Design under Test
HDL	Hardware Description Language
HT	Hardware trojan
IC	Integrated Circuit
IP	Intellectual Property
LFSR	Linear Feedback Shift Register
PDF	Path Delay Fault
RTL	\Register Transfer Logic
S-A-0/1	Stuck-At-1/0
SAF	Stuck At Fault
SoC	System on Chip
SPF	Still Procedure File
STA	Static Timing Analysis

PUBLICATION

- **Rakesh Chanamala**, Sudeendra Kumar and K. K. Mahapatra, "An Improved AES Hardware Trojan Benchmark to Validate Trojan Detection Schemes in an ASIC Design Flow" (Communicated)

1. OVERVIEW

1.1. INTRODUCTION

1.2. MOTIVATION

1.3. RESEARCH OBJECTIVE

1.4. ORGANIZATION OF THE THESIS

1.1. INTRODUCTION:

HARDWARE SECURITY:

Now-a-days due to advancement in technology stealing of confidential information from cryptographic cores has become a very easy task. The technology revolution has fetched various possibilities for these cyber-crimes. An Integrated Circuit manufactured is always desired to be authentic which means it performs only operations originally it is intended to do which is quite a nightmare these days. Integrated Circuits subject to weak design logics and implementation are more vulnerable to these Hardware Security attacks. The adversary can have the control over the IC by introducing a malicious block of additional hardware in the IC that will facilitate the requirement of the adversary. The adversary can introduce Hardware Trojan (HT) at any level of abstraction. The insertion of trojans raised serious concerns regarding possible threats to defence systems. US Defence Science Board reports confirm the malicious insertions into the chips used in military systems [1].

It is very difficult for a designer to design all the modules of his design from the ground level as it is a time consuming process and also the industry may miss a market window due to the delay. Also it is very difficult to afford managing a foundry. Hence the industry is forced to outsource their design for fabrication. Also they purchase IP cores from third party IP vendors. Here comes the loop hole for the intruders to gain access over our design. The IP cores form the vendors may not be trustworthy and are HT inserted.

Detection of these HTs is not an easy task by inspection method as each IC contains billions of transistors which would cost a life time even to count them. Hence the objective of development of various Verification Metrics that improve hardware security has become inevitable. Verification is a pre-silicon process and is to be done prior to fabrication. Some of the verification metrics include Formal Verification, Coverage metrics, Model Checking and

Equivalence checking. Every technique cannot detect all types of HTs. Each technique has its own context of application.

HARDWARE TROJANS:

Hardware Trojans have been gaining a lot of attention in past few years by researchers and governmental organisations as these are mainly built to gain access to secure devices and their data. Hardware Trojans can be inserted into 3PIPs by IP vendors during IP design to steal security information/data from other IPs in the system. Detection of such Trojans is extremely difficult since there is no known golden model for 3PIPs as IP vendors usually provides specification and source code, both of which may contain Trojans. The conventional side-channel techniques for IC trust are not applicable to IP trust. When a Trojan exists in an IP core, all the fabricated ICs will contain Trojans. The only trusted component would be the specification from the SOC designer which defines the function, primary input and output, and other information about the 3PIP that they intend to use in their systems. Automation tools from vendors like Synopsys, Cadence etc, which are generally trusted [2]. Also world's supreme powers rely on defence electronic equipments [3] that needs to be authentic always.

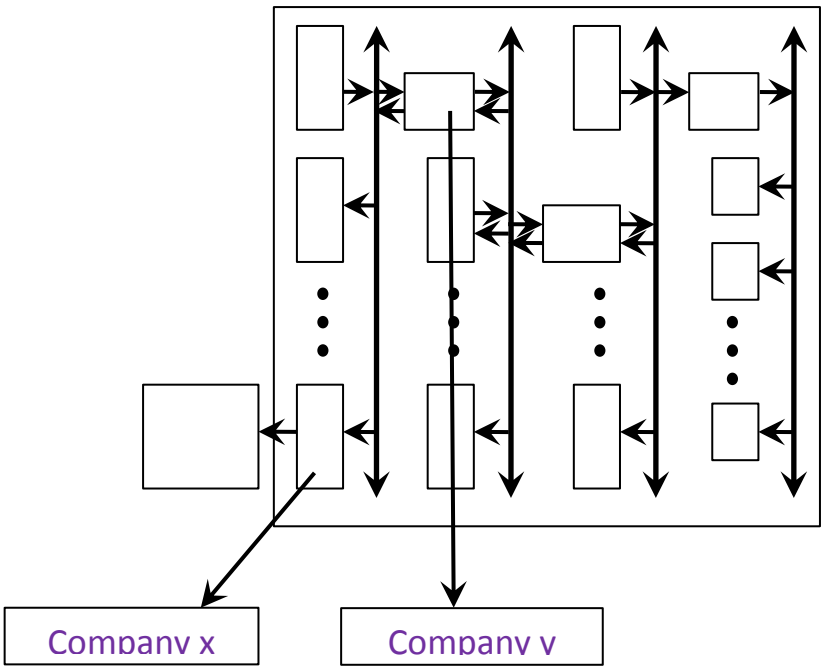

Figure 1-1 Inclusion of 3PIP's (malicious) in SoC

A Trojan can be very well hidden during the normal functional operation of the 3PIP supplied as register transfer level (RTL) code. A large industrial-strength IP core can include thousands of lines of code. Identifying the few lines of RTL code in a soft IP core that represent a Trojan is an extremely challenging task. Cryptographic cores are most vulnerable to the side channel attacks[4].

In fact, Hardware Trojans are alterations to the original circuits, which lets the intruder to access the hardware. These malicious inclusions in a chip can take many forms. It will be easier to identify these malicious inclusions if a proper classification of HTs is done. Hence a comprehensive Taxonomy would increase the effectiveness of application of detection techniques. Hardware Trojans can be easily inserted at RTL stage rather than manufacturing stage[6].

One of the way in which Hardware Trojans can be defined is, “Hardware Trojan (HT) is a malicious modification in the circuitry of an integrated circuit which is completely characterized by its physical representation and its behaviour”.

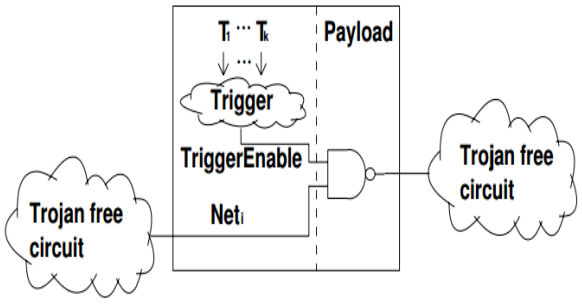

Figure 1-2 Trojan Structure

These Trojans try to bypass or disable the security fence of a system. It may leak valuable and confidential information and sometimes they could destroy the entire chip or components of it.

Depending up on the Physical, Action and Activation Characteristics a detailed flow is presented in the figure 1-3

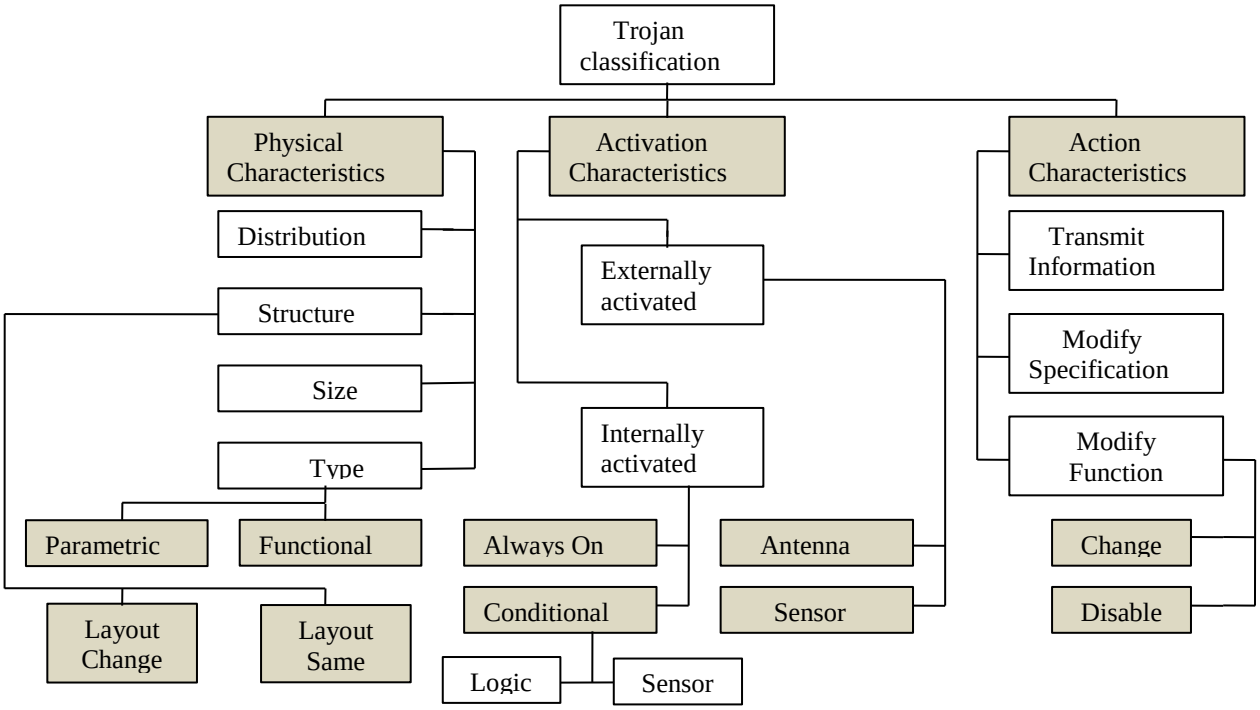

Figure 1-3 Detailed Trojan Taxonomy [2]

Physical Characteristics category describes physical changes that can be manifested in the design. The type category is partitioned in to Parametric and Functional classes. The Functional class depicts the Trojans that are physically introduced by addition or deletion of additional circuitry. The Parametric class depicts the Trojans that are realized by introducing alterations in existing wires and logic. The Distribution class describes about the location of the HT in the physical layout. Sometimes the adversary is forced to regenerate the layout to include Trojan, which is described by the structure category. Any changes in the chips physical design would change the power dissipation and delay characteristics of the chip that would facilitate the detection of its corruption easily[2].

Activation Characteristics describes the criteria or condition on which the Trojan triggers its functionality in payload. These can be externally activated by a sensor or an antenna from outside world or internally activated. The HT triggered internally can be on always and can modify the functionality at any instant. Or they can be activated corresponding to a triggering criterion.

Action Characteristics corresponds to the malicious action of the HT inserted chip. The disruptive action may be like transmitting or leaking information, modifying functionality and specification. They can either modify the functionality or even disable the functionality. Modification of functionality is done either by addition of redundant logic or by bypassing the original logic that directly affects the integrity of the circuit. Some of the instances are like modification of data stored in memory or bypassing a computational operation. Other action characteristic is modification of specification which means the ASIC's parametric properties like clock timing and power are affected. This is achieved by modifying the wires and number of transistors in the design.

IMPORTANCE OF TROJAN BENCHMARKS:

In general Third-Party IP cores are available in three categories namely Hard IP, Firm IP and Soft IP cores. Hard IP cores are defined at physical levels that are in the form of layout or GDSII file format. Firm IP cores are synthesized for specific libraries. Soft IP cores are described in Hardware Description Language (HDL) are very popular due to their flexibility that they can exist both as netlists and HDL code and hence it provides greater flexibility for the adversary to include malicious behaviour at this level. Also inserting a code doesn't cost but modifying or adding additional circuitry to Hard IP however is an economic issue. As the Industry standard IP cores comprises of thousands of lines of HDL code it is very difficult to

detect and small alteration or inclusion of a malicious behaviour. Hence Soft IP cores are the means by which the attackers make use to serve his purpose.

Analysis of Trojan inserted Benchmark designs has drawn much attention. These Trojan benchmarks will let us familiar with their physical, action and activation characteristics of Hardware Trojans. As it is evident that not all types of Hardware Trojans can be detected by a single technique, using these Trojan inserted Benchmarks a comprehensive flow of detection techniques, that describes the application of a particular technique on a specific design at a specific instant, can be developed. Hence design, analysis and study of various Hardware Trojan Benchmarks have gained attention of verification engineers.

1.2. MOTIVATION*:*

Due to globalization, various stages of System on Chip (SoC) development has taken a new face where the there is a flexibility for the industry to opt different levels of abstractions from different sources. This has made the SoC development comparatively a less time consuming process than earlier.

This has somehow made the job easy and difficult as well. Outsourcing of the design for fabrication or verification or testing may cost the loss of authenticity of the design. For instance, a typical SoC organisation is shown in the figure. Not all the blocks are designed by the same company as it leads to missing of a market window where they may run out of time and loose the current existing technology. Hence they opt for Intellectual Property (IP) cores from third-party IP vendors. Here comes the problem about the authenticity and trust-worthiness of the IP cores. The 3-PIP vendor may not supply us with a trust-worthy IP core and these IP cores are to be subjected for various verification techniques to trace out Hardware Trojans inserted, if any. This has motivated for the current study.

1.3. RESEARCH OBJECTIVE:

The objective of this dissertation is study various classes of hardware Trojans nad investigate various Hardware Trojan Detection Techniques available and propose a new comprehensive approach using the available techniques. To implement the idea, various AES Benchmarks from Trust-hub were taken and analysed to implement various Hardware Detection Techniques. Also the strengths and weaknesses associated with them were analysed and a novel benchmark design is proposed to overcome their weaknesses. Synopsys DC™ 2008 version[20] is used for entire research.

ORGANIZATION OF THE THESIS:

The thesis organization flow is listed below

- **Chapter 2:** This chapter focuses on various simulation-based Hardware Trojan detection techniques using verification and corresponding literature survey.
- **Chapter 3:** This chapter focuses on detailed description of Advanced Encryption Standard (AES) and RSA Benchmark designs.
- **Chapter 4:** This chapter details strengths and weaknesses of AES and RSA benchmarks.
- **Chapter 5:** This chapter proposes Novel AES Trojan benchmark designs of whose malicious behaviour and the cause is very difficult to detect using existing verification techniques.
- **Chapter 6:** This chapter describes possible future work and conclusion of work done.

2. VERIFICATION METRICS FOR TROJAN DETECTION

2.1. SYSTEM ON CHIP VERIFICATION FOR TROJAN DETECTION

2.2. VERIFICATION v/s TEST

2.3. VERIFICATION METRICS

2.3.1. FORMAL VERIFICATION

2.3.2. COVERAGE METRICS

2.4. HARDWARE TROJAN DETECTION PROBLEMS AND SOLUTIONS

2.1. SYSTEM ON CHIP VERIFICATION FOR TROJAN DETECTION:

There is no "Golden bullet" for detecting all classes of HTs. A specific technique should be incorporated for detection of specific class of HTs. Detection prior to fabrication at RTL level could save significant time and man power.

Verification metrics are used in general to verify the correctness of the design prior to fabrication. It is a challenging task for the verification engineer to detect the intruder included malicious behaviour.

2.2. VERIFICATION v/s TEST:

Verification and Testing are equally important in a System-on-Chip (SoC) design as they both contribute to the flaw less design both in physical and functional. However both have significant differences like verification is an analytic approach to verify whether the functionality of the design is matching with the specification or not. Unlike verification testing ensures the fabricated device from the netlists is physically flawless. Various differences between verification and testing are shown in Table 1-1.

Table 2-1 Verification versus test

VERIFICATION	TESTING
1. It accounts for the correctness of the design.	1. It accounts for manufacturing flaws associated with physical design.
2. It is a pre-silicon process	2. It is a post-silicon process
3. It accounts for quality of design prior to manufacture	3. It accounts for quality of device
4. It is exercised only once on the design	4. Each IC fabricated has to undergo testing.

2.3. VERIFICATION METRICS:

Various Simulation-based verification metrics include Formal Verification and Functional Verification.

2.3.1. FORMAL VERIFICATION:

Formal verification is an approach by which correctness of various properties is verified. Formal verification is categorised into Model Checking and Equivalence checking.

MODEL CHECKING AND EQUIVALENCE CHECKING:

Model Checking:

Model Checking is done by writing the specification of design as temporal logic and verifying them with the design.

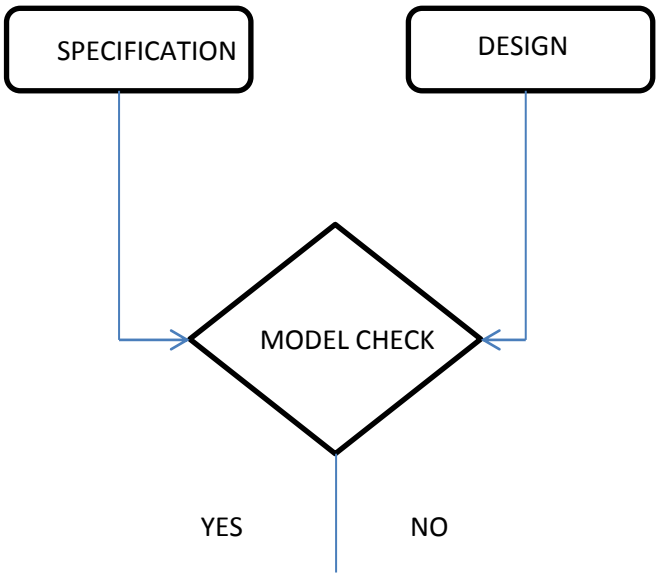

Figure 2-2-1 Model Check block diagram

The Figure represents the model checking for specification against its design. If Model check passes then it implies that the specification versus design match passed. In case, if the model check fails then the design is to be redesigned to ensure security [9][17].

Equivalence Checking:

Equivalence checking verifies the equivalence between RTL design and netlists to ensure whether the functionality and timing of RTL design is matching with its netlists or not. This is illustrated in the figure.

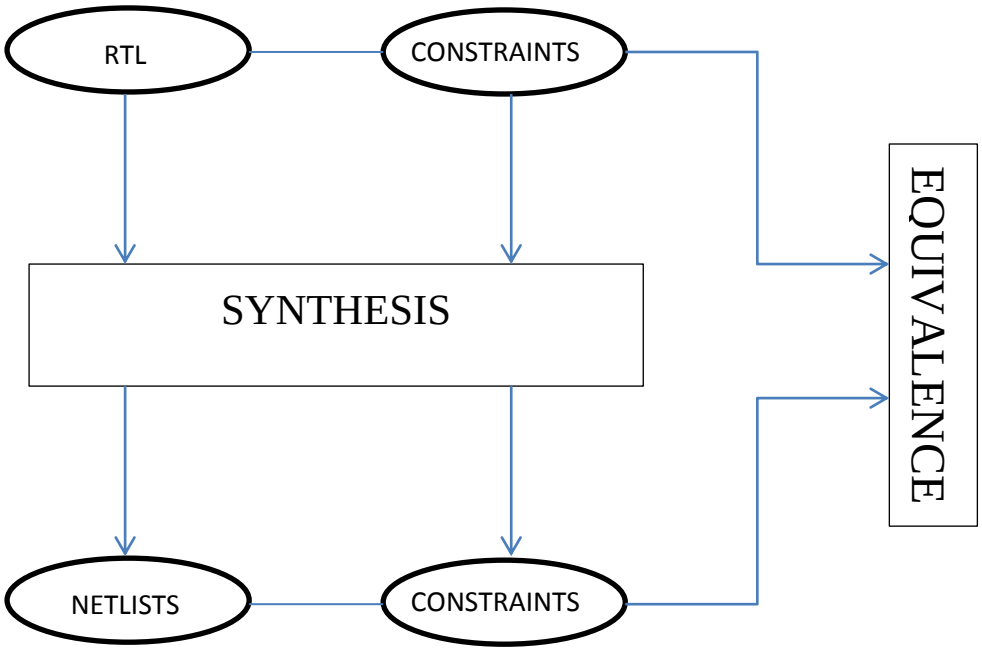

Figure 2-2 Equivalence Check block diagram

The Figure represents Equivalence checking on RTL versus netlists. If the Equivalence test passes there is no issue if it fails the RTL and netlists should be re-examined for mismatch.

2.3.2. COVERAGE METRICS:

Coverage Metrics are those by which means a RTL design is verified. These metrics can ensure the correctness of the design with respect to functionality and also contribute for detecting malicious blocks, if any.

Coverage metrics are broadly classified into

- Code Coverage
- Functional Coverage

CODE COVERAGE:

Code Coverage will report the effectiveness of the test-bench. In general none can write in hand manually all the possible input combinations for inputs whose widths are larger. Code coverage will report what parts of the design are covered and left uncovered by the test-bench. This feature can help the verification Engineers in detecting Hardware Trojans. Code Coverage is further sub divided into

- Line Coverage
- Statement Coverage
- Condition Coverage
- Block Coverage
- Expression Coverage
- FSM Coverage
- Toggle Coverage

Out of all Toggle coverage and FSM coverage has gained Prominence with respect to the area of Hardware Security as the pin associated with malicious block will not toggle until the Trojan triggering condition occurs and this can be dealt using Toggle Coverage Report. Also the malicious block will try to bypass the original FSM states which get caught in FSM coverage.

FUNCTIONAL COVERAGE:

A simple test-bench written in Verilog coding style cannot detect the functional flaws in the design on which the test-bench is intended to test for. This is because the test bench may or may not correspond to the inputs that trigger the unwanted behaviour. Hence their functionality should be tested by writing a powerful test-bench where the inputs corresponding to the malicious trigger condition is necessary.

Functional Coverage is further categorised into

- Control Oriented Functional coverage
- Data Oriented Functional Coverage

CONTROL ORIENTED FUNCTIONAL COVERAGE:

Control Oriented Functional Coverage is one of the verification metrics based on assertions. The design behaviour's temporal aspect can be verified using this. Assertions are written in System Verilog Language defining the behaviour of the control pins, the coverage can be achieved.

It plays vital role in identifying the functional flaws very easily compared to data oriented coverage using minimum lines.

DATA ORIENTED FUNCTIONAL COVERAGE:

Data oriented Functional Coverage deals with specific values, transitions, or combination of both that the variable receives. It keeps track of the data throughout the simulation.

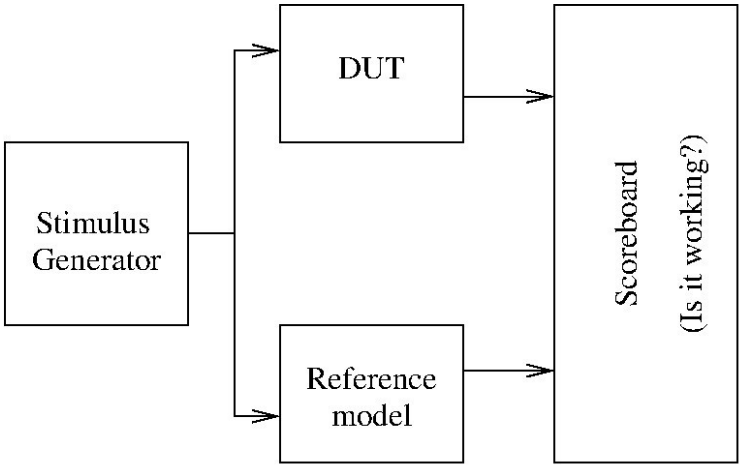

Figure 2-3 Data Oriented Coverage Block Diagram

In this the exact values that the variable is supposed to receive are taken in a bin and the variable is subjected to verification for the values is actually receives versus the values it should actually receive (values stored in bins).

In the figure 2-3 the reference model actually holds the bins that store the values that a variable likely to hit for and the DUT is verified against these values. The scoreboard record corresponding score. The score is recorded as a *hit* and *mis-hit* count. Hit count corresponds to the matched data values of the data values occurred against the data values in bins. *mis-hit* corresponds to the mismatched value

2.4. HARDWARE TROJAN DETECTION PROBLEMS AND SOLUTIONS:

There are already existing comprehensive flow of Trojan detection techniques for various classes of Trojans [7][8]. Hardware Trojan Detection techniques use verification, testing, timing fingerprints and side-channel analysis [9] [10] [11]. Detecting Hardware Trojans in third party IP core using conventional verification techniques is not as easy as it is said. Xuehui Zhang and Mohammad Tehranipoor of University of Connecticut in their paper, case study: Detecting Hardware Trojans in Third-Party Digital IP Cores have presented a study on difficulties associated with Trojan detection. They proposed a detection flow in

which they used model checking and equivalence checking in their techniques to reduce the suspicious signals in RS232 Trojan benchmark designs[13].

According to them the detection is carried out in two phases where in the first phase the third party IP is subjected to formal verification and coverage analysis. They used around five different test-benches to conduct coverage analysis. If formal verification and coverage analysis fails to detect the inserted Trojans then the design is tested with test patterns generated by ATPG. If the coverage appears to be 100 %, then it is Trojan free otherwise it is considered as Trojan inserted. These steps identify the suspicious signals.

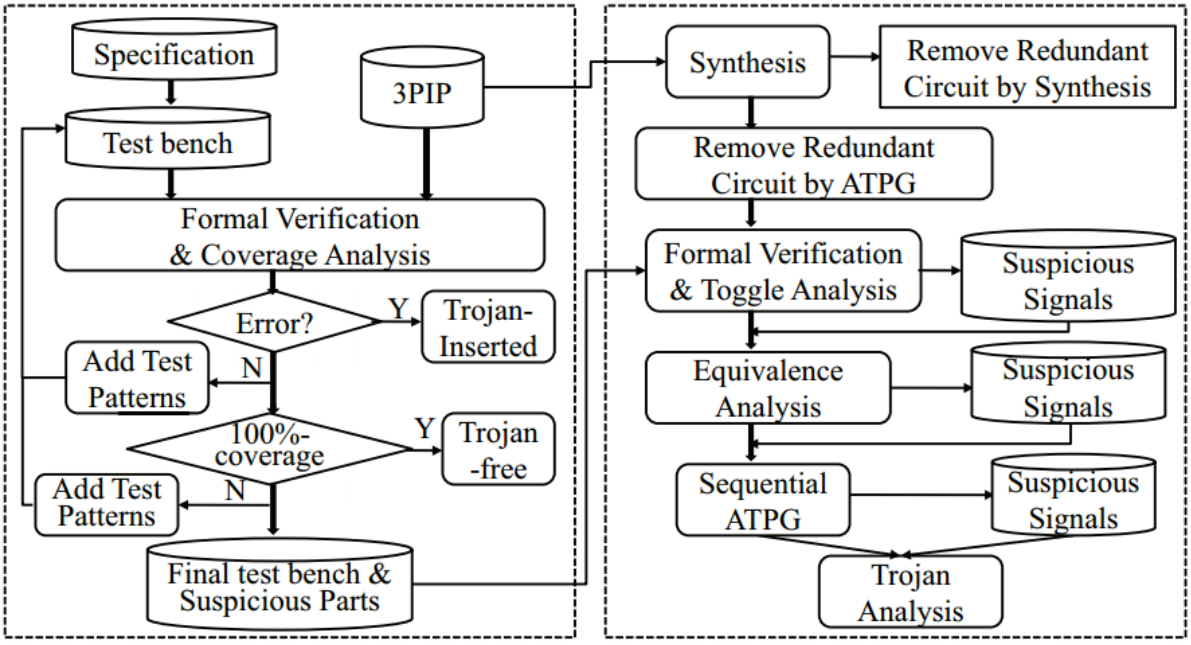

Figure 2-4 Test-bench generation, suspicious signals identification and analysis flow [T]

Second phase is analysis of suspicious signals. In this phase they tried to remove the redundant circuitry by equivalence checking theorems and testing the design for untestable stuck-at-faults using sequential ATPG. If stuck-at-faults are untestable implies that

the outputs corresponding to the design with 100 % coverage will match with the one with less than 100 % coverage.

Trey Reece and William H. Robinson in their paper Analysis of Data-Leak Hardware Trojans in AES Cryptographic Circuits a detailed study to demonstrate the impact of the Trojans inserted on area and Leakage power.

They have taken 21 AES Trojan inserted Benchmarks from trust-hub[12] repository to carry out Area and Power Analysis on them. They mentioned that the area footprint is in the range of -6,018 to 6,506 square micrometres. Out of the 21 AES benchmarks they reported only half of them increased the area that too not much significantly, which means that the impact of the Hardware Trojans inserted on the area is less significant.

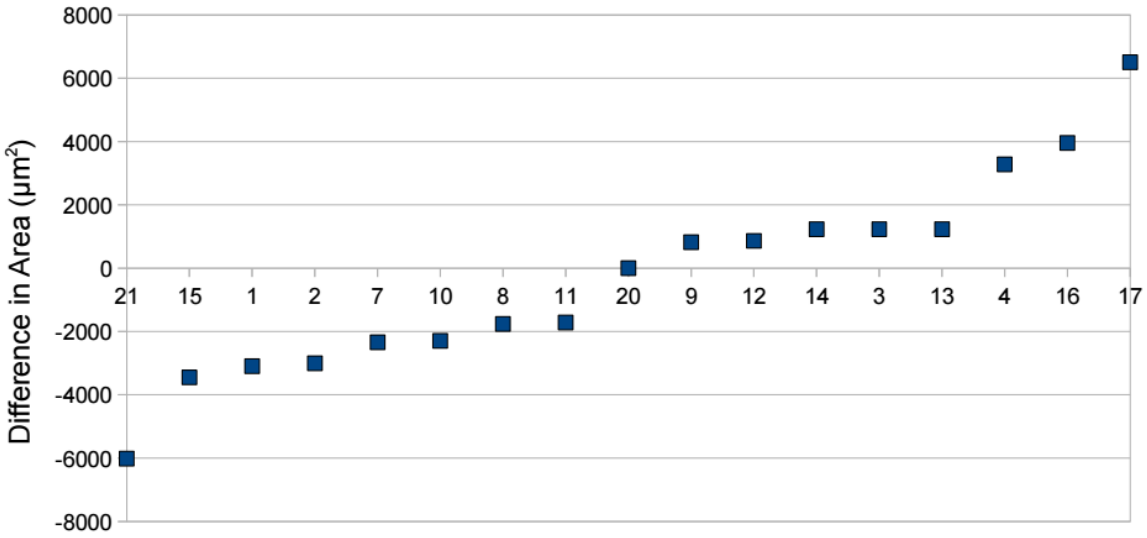

Figure 2-5 Difference in areas of AES Benchmarks [14]

The numbers on the x-axis indicate the benchmark number. Out of all AES benchmarks they reported that AES-T2100 has highest negative difference and AES-T1700 has highest positive difference. According to their work AES-T600 benchmark has literally reported the same area as that of the Trojan free benchmark. This makes the approach of

considering the area overhead for Trojan detection very uncertain as the impact of the redundant malicious block over the area is insignificant.

Also they provided a detailed study on leakage power of twenty one AES benchmarks. The details regarding the power values are shown in the figure. The values reported for the leakage power analysis didn't report any even spread like area analysis. According to them the maximum obtained leakage power footprint is 47.5 microwatts and the minimum obtained footprint is of 6.9 microwatts.

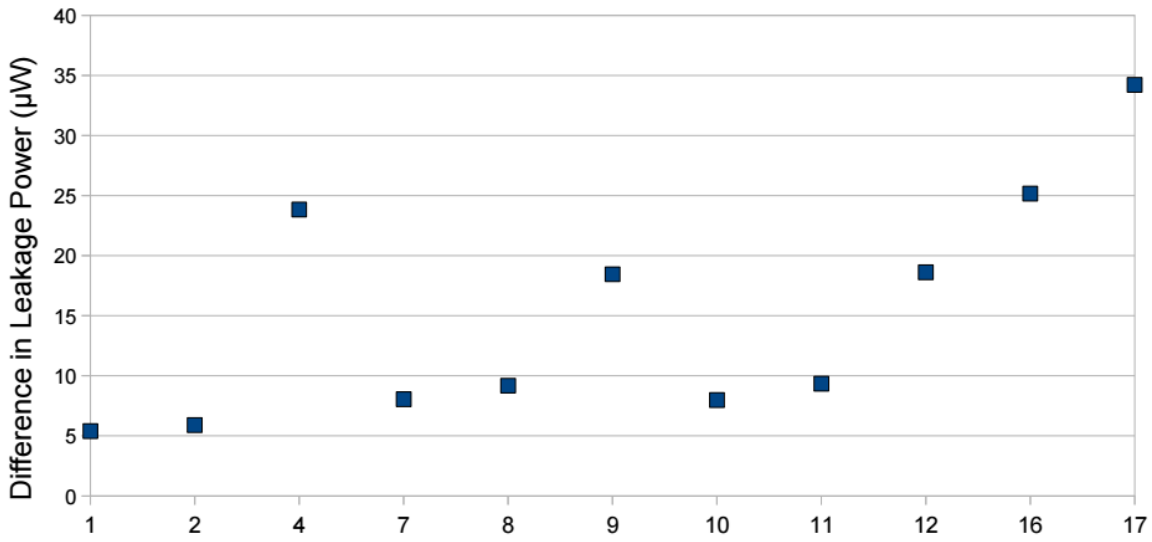

Figure 2-6 Difference in leakage power of AES Benchmarks [14]

Michael Bilzor, Ted Huffmire, Cynthia Irvine, and Tim Levin in their paper evaluating Security Requirements in a General-Purpose Processor by Combining Assertion Checkers with Code Coverage they enhanced an existing method in order to create dynamic checkers based on assertions[9]. Also they developed a checker-generator tool called psl2hdl that converts Property Specification Language to Hardware Description Language. However verification engineers are already using this assertion based functional check, these authors extended their work for general-purpose processors

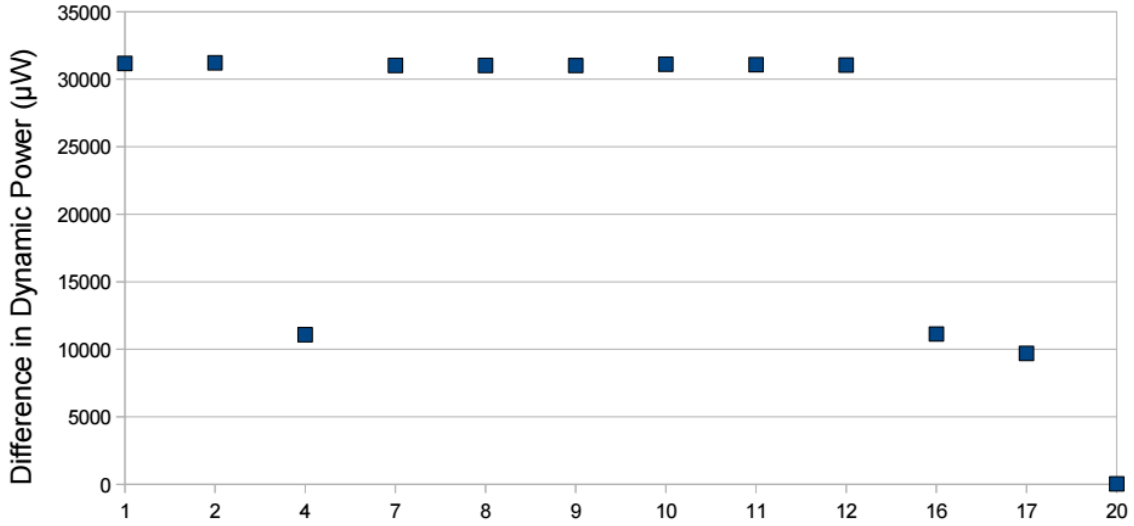

Figure 2-7 Difference in Dynamic power of AES Benchmarks [14]

.

Yier Jin and Yiorgos Makris in their work presented a Hardware Trojan detection technique based on path-delay finger printing. They tried to validate the authenticity of any design based on their delay finger prints[6]. Also in their work they demonstrated Combinational Trojan architecture, Sequential Trojan architecture and existing difference between them.

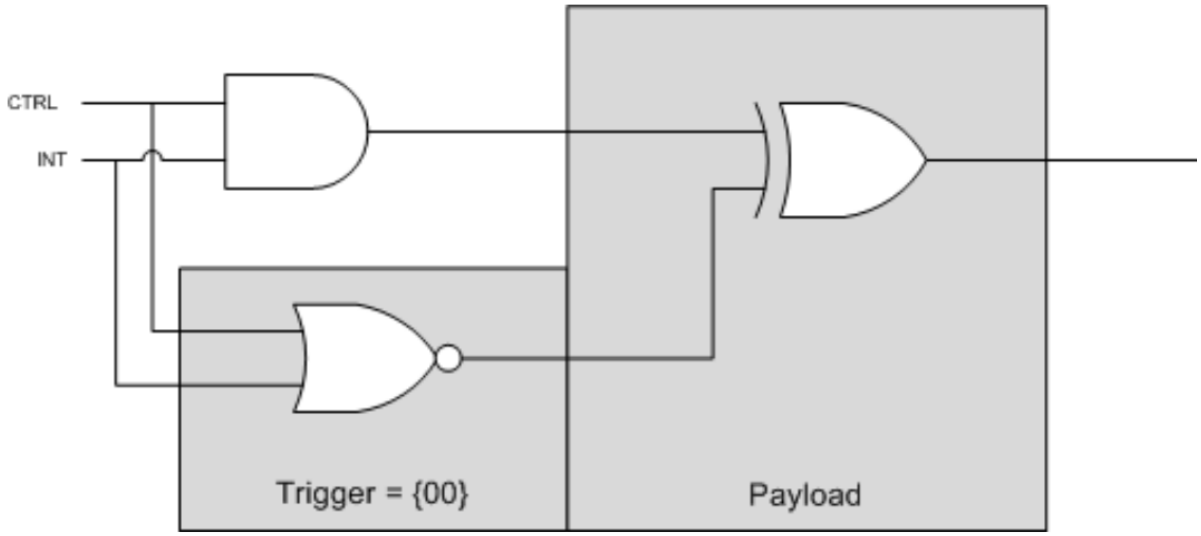

Figure 2-8 Architecture of Combinational Trojan [6]

In the figure NOR-gate is redundant corresponding to the trigger condition {00}. The existence of gate doesn't change the output corresponding to the triggered case. The payload monitors the control signal (CTRL) and interrupt (INT) signals and gets activated when the control signal (CTRL) and interrupt (INT) is at low voltage levels. Combinational Trojan design comprises of only combinational gates. Unlike Combinational Trojans, sequential Trojans do have at least one memory element.

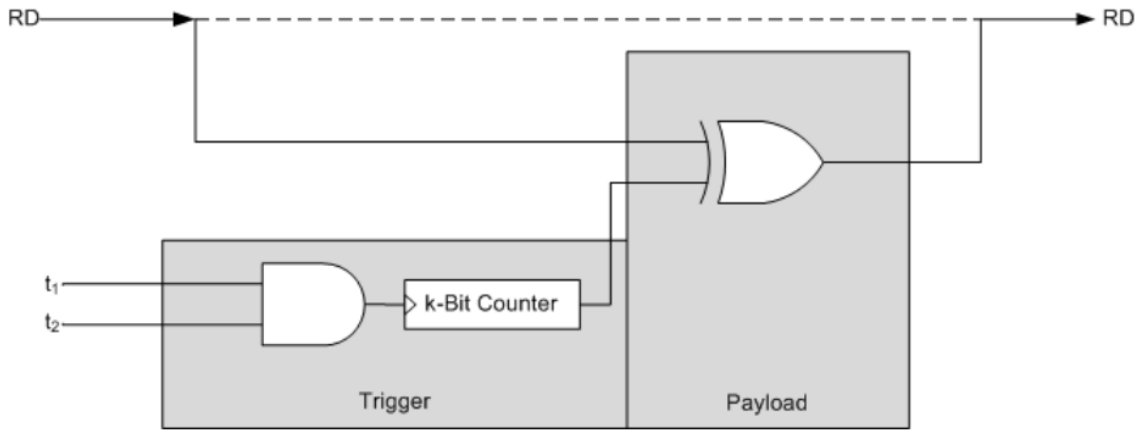

Figure 2-9 Architecture of Sequential Trojan [6]

In the figure the malicious block is a k-bit counter that doesn't cause any functional flaw but manages to consume additional power than its original specification. They used Static timing Analysis tool to get the finger prints. However inclusion of malicious blocks in any design will affect the parametric specifications like delay and power. They key idea here is to take the path delay finger prints of the Golden (Trojan-Free) design and compare them with the Trojan inserted designs.

3. THE AES AND RSA ALGORITHMS

3.1. AES

3.2. RSA

3.1. ADVANCED ENCRYPTION STANDARD (AES):

Advanced Encryption Standard (AES) is a cryptographic security algorithm developed by National Institute of Standard and Technology (NIST) of United States in the year 2001 [wiki]. The AES algorithm encrypts electronic data into cipher data which again can be decrypted using the same key. The basic block diagram of AES is shown in fig:

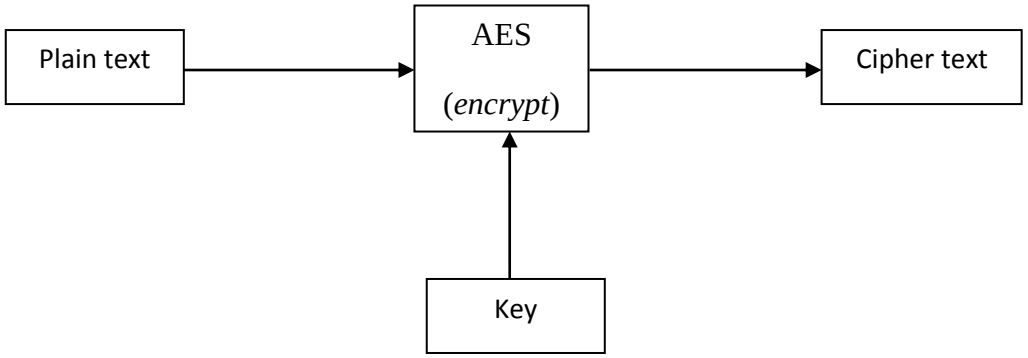

Figure 3-3-1 AES Encryption Block Diagram

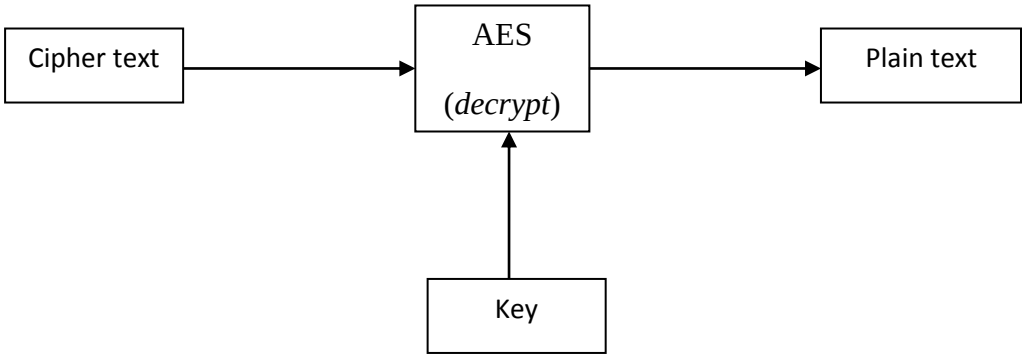

Figure 3-3-2 AES Decryption Block Diagram

The key size can be 128-bit, 192-bit or 256-bit and the block size is fixed of size 128-bit. AES works on the principle of substitution-permutation network [wiki]. Both encryption and decryption uses the same key.

STRUCTURE OF AES ALGORITHM:

AES encryption is done in ten rounds if the key size is of 128-bit, 12 rounds if the key size is about 192-bit and 14 rounds if the key is about 256-bit. All rounds are identical except the last one where the key addition is done. Various steps of AES algorithm are shown in the figure.

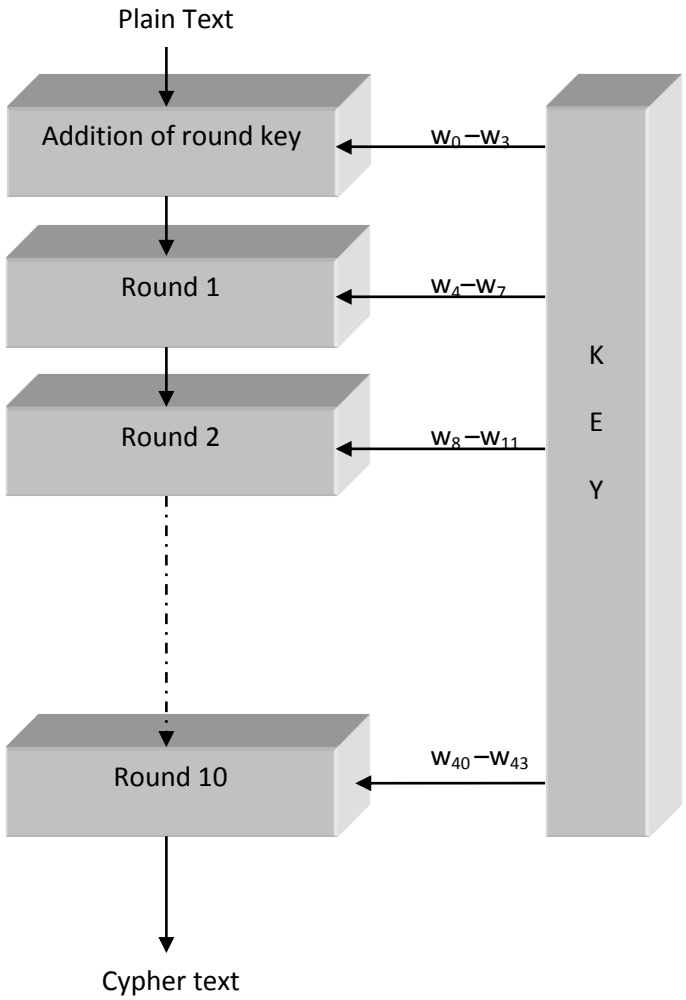

Figure 3-3-3 AES encryption for 128-bit plain text

Figure 3.3 depicts the overall structure of AES algorithm. The 128-bit key is expanded and the first four words of the key schedule are XORed with the input array before the start of round-processing. Also during decryption the same thing happens, that is the

cypher text is XORed with the last four words of the key schedule. Key scheduling is based on an algorithm described in the later sections.

ONE ROUND:

Each Round is comprised of four operations namely

- Byte Substitution
- Shift Rows
- Mix Columns
- Addition of Round Key

The overall structure of each round is shown in figure

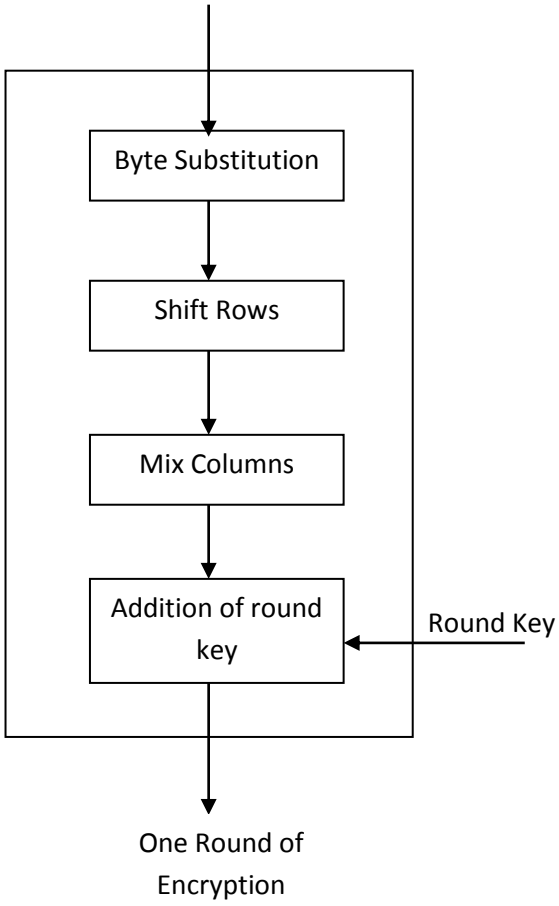

Figure 3-4 various steps in one Round of AES[21]

BYTE SUBSTITUTION:

Each byte of the plain text is substituted with another byte from a look up table of size 16 X 16. In order to determine the substitute byte each input byte is divided into two 4-bit streams corresponding to the integers 0 to 15 and equivalent hexadecimal of 0 to F. Out of the two hexa-decimal values of the byte one is referenced as a row index and the other as column index. The look up table is constructed using GF(2^8) arithmetic followed by bit scrambling. Byte Substitution step reduces the correlation between input plain text and cypher text.

SHIFT ROWS LAYER:

The shift rows layer do not shift the first row of the state array, circularly shifts by one byte of the second row to left, third row two bytes to left and fourth row three bytes to left.

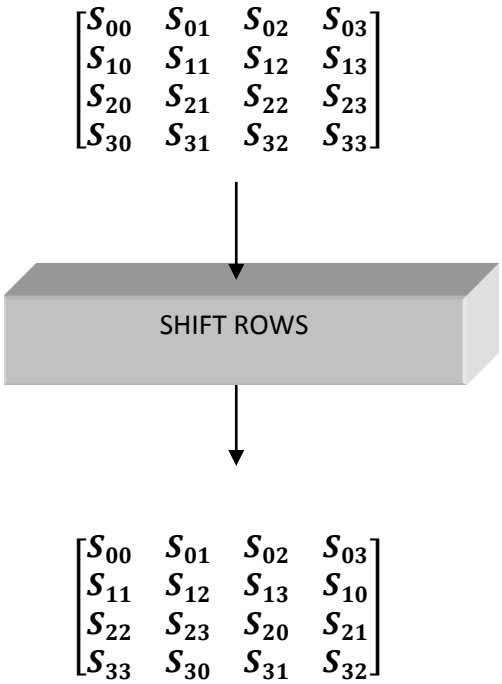

Figure 3-3-5 State vector matrix before and after shift rows operation

Here the first column of the state vector matrix is the first four bytes of the input plain text and thus the result of this shift rows operation will scramble the order of the input block Bytes.

MIX COLUMNS:

In this layer all the bytes of the columns will be substituted by corresponding bytes which are function of all bytes in the column. To be specific the function is the sum of double the byte, triple the next byte, the very next byte and the following byte. Mix column operation is depicted in the fig

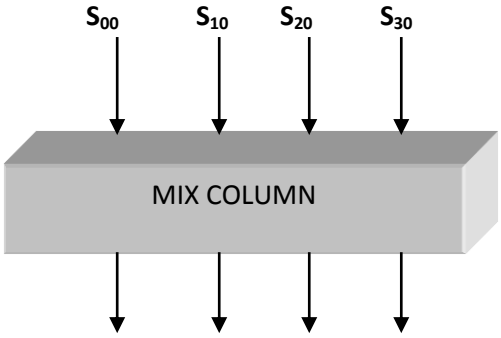

Figure 3-3-6 Mix Column layer

The Corresponding transformation for encryption is given by

$$\begin{bmatrix} C_{00} \\ C_{10} \\ C_{20} \\ C_{30} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{00} \\ S_{10} \\ S_{20} \\ S_{30} \end{bmatrix}$$

KEY EXPANSION:

The main idea behind key expansion is that a change in one bit of the key would affect the round key for many rounds. During encryption or decryption the original key is not directly given during each round but a 128-bit round key obtained from the original

key on its expansion is used at each round. In this process the state vector is XORed with the round key once for each round.

$W_{0,}$ $W_{1,}$ W_2 and W_4 are words of each 64-bit where W_0 is obtained from first column (first four bytes) of the state vector matrix, W_1 is obtained from the second column (next four bytes) of the state vector matrix and so on.

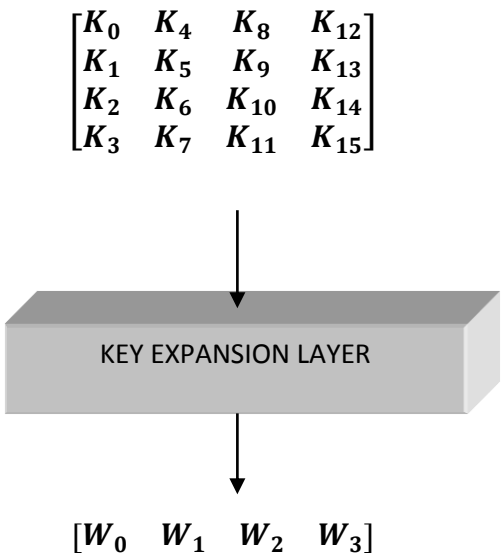

Figure 3-3-7 Key Expansion

These are XORed bitwise with the input prior to round based processing. $W_{0,}$ $W_{1,}$ W_2 and W_4 are expanded further into a 44-word key schedule each of 64-bit and each of the four subsequent words were applied to each round during key scheduling[21]. The key expansion process is illustrated in the figure.

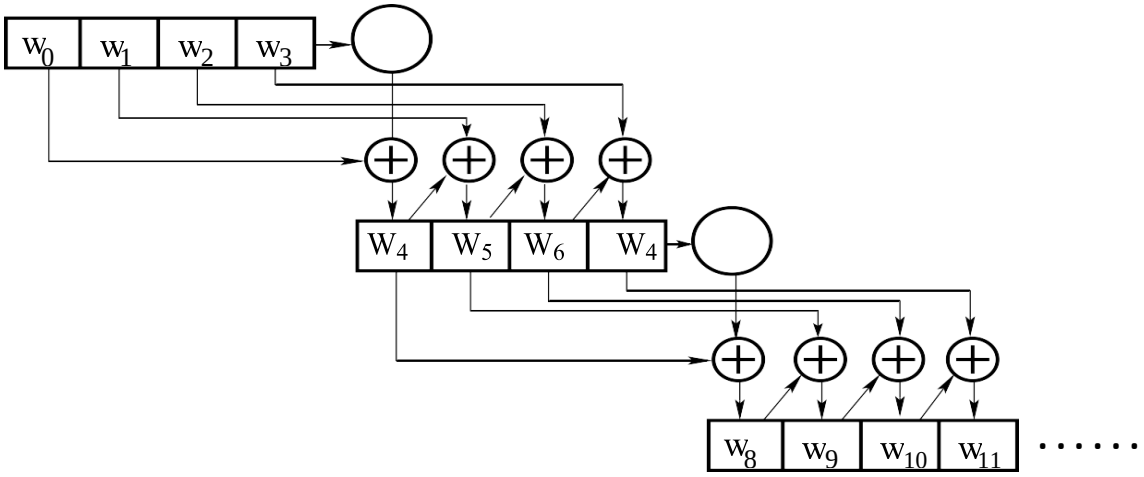

Figure 3-3-8 The Key Expansion Flow [21]

Except the first word in the new grouping, every other word is the result of XOR operation of its previous word with the corresponding word in the previous grouping.

3.2. RSA*:*

RSA is a popular public-key cryptosystems that ensures secure data transmission. It is named after the names of its designers, Ron Rivest, Adi Shamir, and Leonard Adleman. RSA was first published in the year 1977.

Unlike AES, RSA Cryptosystem is asymmetric nature which means that encryption and decryption is not dictated by same key. Encryption is done using public key and decryption is done using a private key. The data is secured as long as the private key will be kept secret. The symmetric and asymmetric cryptosystems are depicted in the figure.

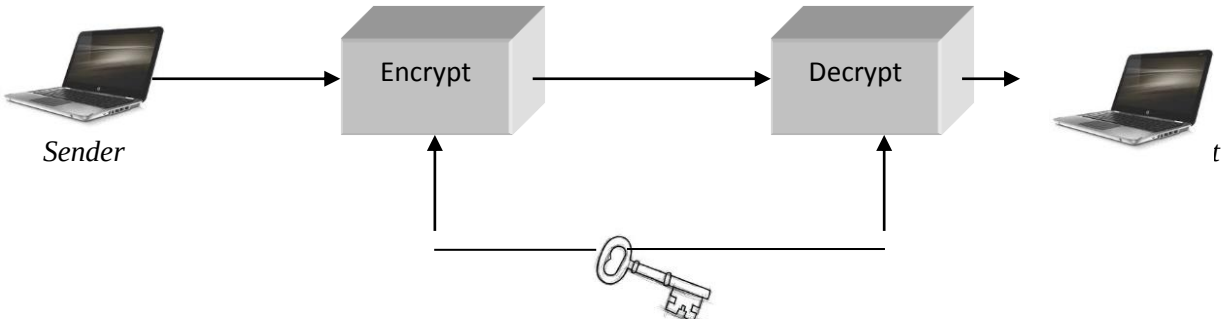

Figure 3-9 Symmetric Cryptosystem (AES)

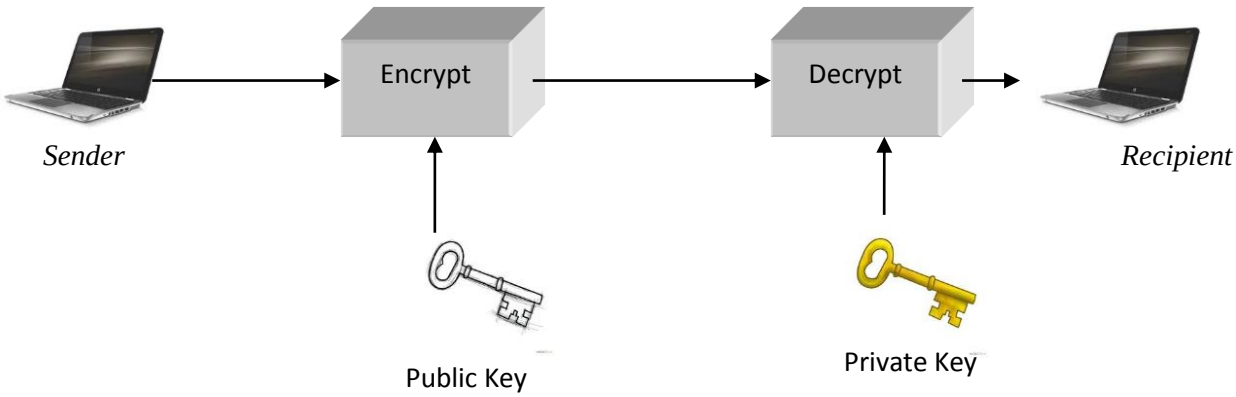

Figure 3-10 Asymmetric Cryptosystem (RSA)

Figure illustrates the encryption-decryption process in symmetric and asymmetric cryptosystems.Public key can be shared with anyone but private key is shared only with the reciepent for whom data transmisiion is intended for.

Various steps in RSA Algorithms are Key generation, Encryption and Decryption.

KEY GENERATION:

Unlike symmetrical cryptosystems like AES, asymmetrical cryptosystems like RSA require computation of a pair of keys namely Public Key (K_{pub}) and Private Key (K_{pr}). Public Key and Private Key generation steps are as listed below.

- Choose large distinct primes p and q (Prime Numbers are only chosen because the can ensure randomness required for security purposes).
- Evaluate $n = p.q$
- Evaluate $\emptyset(n) = \emptyset(p).\emptyset(q)$, where $\emptyset(p) = p - 1$ and $\emptyset(q) = q - 1$
- Choose K_{pub}= $e \in \{1,2,\ldots. \emptyset(n-1)\}$, such that $GCD[e, \emptyset(n)]=1$
- Evaluate $K_{pr}=d$ such that $d.e=1 \bmod \emptyset(n)$. Here d is the private key.

ENCRYPTION:

If a *person-A* shares public key (*n*,*e*) with *person-B* and with private key *d* held secret. *person-B* sends message X to *person-A* as described below.

X is to be converted into an integer *x*, such that $0 \leq x <$ n and gcd(*x*, *n*) = 1. Then the encrypted cypher text is given by the equation

$$Y = E_{Kpub}(X) = X^{e} \bmod n$$

Where Y is the cypher text and E_{Kpub} represents encryption function.

DECRYPTION:

Given Private Key (K_{pr}) = d, the decrypted plain text is given by the equation

$$X=D_{Kpr}(Y) = Y^{d} \ mod\ n$$

4. TROJAN BENCHMARKS AND THEIR STRENGTHS AND WEAKNESSES

4.1. AES TROJAN BENCHMARKS

4.2. POWER AND AREA ANALYSIS

4.3. STRENGTHS AND WEAKNESSES OF AES BENCHMARKS

4.4. RSA BENCHMARKS

4.5. STRENGTHS AND WEAKNESSES OF RSA BENCHMARKS

4.1. AES TROJAN BENCHMARKS:

Twenty one AES Trojan Benchmarks with various malicious behavior are taken from the trust-hub for analysis and detection of Trojans inserted in them using existing conventional Hardware Trojan detection techniques. The description of twenty one AES benchmarks is tabulated in the table.

Table 4-1 AES Benchmarks Description

Benchmark	Effect	Side-channel	Description
AES-T100	leak info	Power	The Trojan use pseudo-random number generator (PRNG) to create a CDMA code sequence. CDMA sequence is forwarded to a leakage circuit to set up a covert power side-channel. PRNG initialized to pre-defined value. (Always on Trojan)
AES-T200	leak info	Power	Same as AES-T100 but, PRNG initialized to pre-defined text. (Always on Trojan).
AES-T300	leak info	power	Trojan leaks a byte of the round key for each round of the key schedule. The leakage circuit is a shift register and loaded with alternating zeros and ones when Trojan is inactive. When Trojan leaks key, it results in additional dynamic power consumption. (Always on Trojan).
AES-T400	leak info	RF- signal	Modulating an unused pin on a chip generates an RF signal to transmit the key bits. Trojan gets activated with the predefined input plaintext.
AES-T500	DoS	NA	Trojan is triggered by pre-defined input and drains battery within a short duration.
AES-T600	leak info	Leakage current	When a specific plaintext is given as input, the Trojan leaks the secret key through the leakage current. The leakage circuit consists of a shift register holding the secret key.
AES-T700	leak info	Power	Same as AES-T100 but, Trojan trigger when pre-defined input is observed.
AES-T800	leak info	Power	Same as AES-T100 but, Trojan trigger when pre-defined input is observed.
AES-T900	leak info	Power	Same as AES-T100 but, Trojan trigger after 2^{128} encryptions.
AES-T1000	leak info	Power	Same as AES-T100 but, Trojan trigger when pre-defined input is observed.
AES-T1100	leak info	Power	Same as AES-T100 but, Trojan trigger when pre-defined input is observed.
AES-T1200	leak info	Power	Same as AES-T100 but, Trojan trigger after 2^{128} encryptions.
AES-T1300	leak info	Power	Trojan is triggered by specific input, it leaks key through increase in dynamic power.
AES-T1400	leak info	Power	Trojan is triggered by input of specific sequence, it leaks key by increase in dynamic power.
AES-T1500	leak info	Power	Trojan is triggered after 2^{128} encryptions, leak key through an increase in dynamic power.
AES-T1600	leak info	RF-Radio signal	Same as AES-T400 but, Trojan trigger when pre-defined input is observed.
AES-T1700	leak info	RF-Radio signal	Same as AES-T400 but, Trojan trigger after 2^{128} encryptions
AES-T1800	DoS	NA	Trojan is triggered by pre-defined input and drains battery within a short duration.
AES-T1900	DoS	NA	Trojan is triggered after 2^{128} encryptions and drains battery within a short duration.
AES-T2000	leak info	Leakage current	Trojan is triggered by pre-defined input and increases the leakage current.
AES-T2100	leak info	Leakage current	Trojan is triggered after pre-defined number of encryptions and increases leakage current.

RF-Radio Frequency; DoS-Denial of Service; NA- Not Available

From the table it is evident that the malicious behavior of the Benchmarks is either they leak the secret key or perform Denial of Service operation. Here the crypto core is actually AES integrated with a Trojan Side Channel (TSC) [6] that performs the malicious operation. The Trojan inserted AES module would likely appear as shown in figure.

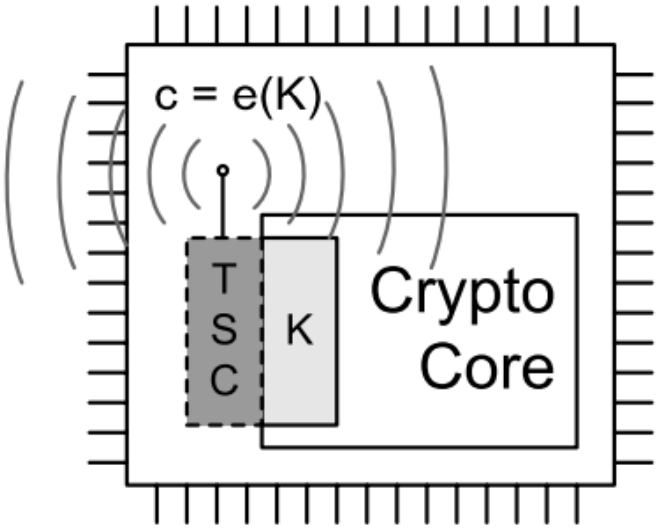

Figure 4-1 Trojan inserted AES [lin]

L. Lin, M. Kasper, T. Güneysu, C. Paar and W. Burleson described the class of TSC introduced performs a known operation on the key so that the function is again inverted and applied to retrieve the secret key. The operations are depicted in the equations below

$$C= e (K)$$

$$e^{-1} (K) =C$$

Where K is the key, C is the output of the Trojan Side Channel block and e is the malicious function that performs the malicious action. As the attacker is the designer of the core, he can easily retrieve back the original key by inverting the malicious function. They modelled the function e to model various Trojans.

Depending on the TSC function the benchmarks are categorized as follows

- Those that perform Key leakage based on spread spectrum technique.
- Those that perform Key leakage based on RF-Radio Transmission using Amplitude Modulation (AM).
- Those that perform Denial of Service (DoS) operation using Shift register based rotation logic.
- Those that perform Denial of Service (DOS) operation using a series of inverters that logic.

CDMA based TSC:

The TSCs employ spread spectrum technique for key leakage. Using Code Division Multiple Access (CDMA) scheme, the leakage of single bits is distributed over many clock cycles.

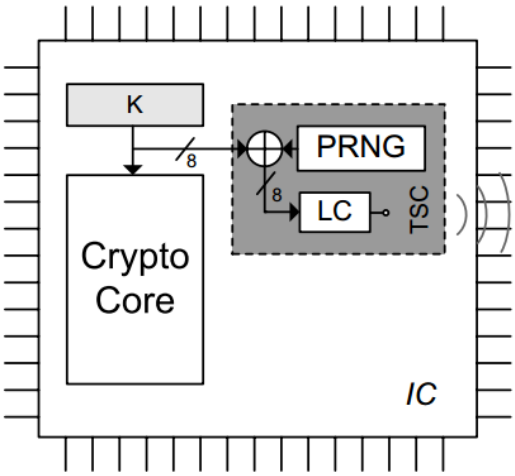

Figure 4-2 CDMA based TSC implementation [lin]

CDMA uses a Pseudo Random Number Generator (PRNG) to modulate the information. Likewise the TSC here uses an LFSR (Linear Feedback Shift Register) which is a PRNG to modulate the key bits. In specific the key bits are XORed with the secret key.

Then the XOR modulated sequence is then transmitted to LC circuit so that a covert CDMA channel is set up in the power side channel. Since that attacker has the knowledge about the PRNG, the covert channel can be distinguished from the noise. The attacker decodes the covert channel by performing correlation demodulation technique. Among the twenty one benchmarks, AES-T100, AES-T200, AES-T700, AES-T800, AES-T900 AES-T1000, AES-T1100 and AES-T1200 possesses the TSC that falls in this category[18][19]. In AES-T100 the Trojan has been always active. In AES-T200 the Trojan is triggered up on the low reset. In remaining the Trojan gets triggered based on a criterion. The criterion may be either an encounter of a particular state or a sequence of states or after a particular number of encryptions.

RF Transmission based TSC:

An unused pin on the chip is modulated to generate RF signal. Unused pin can be used as an RF transmitter to transmit the information to the adversary

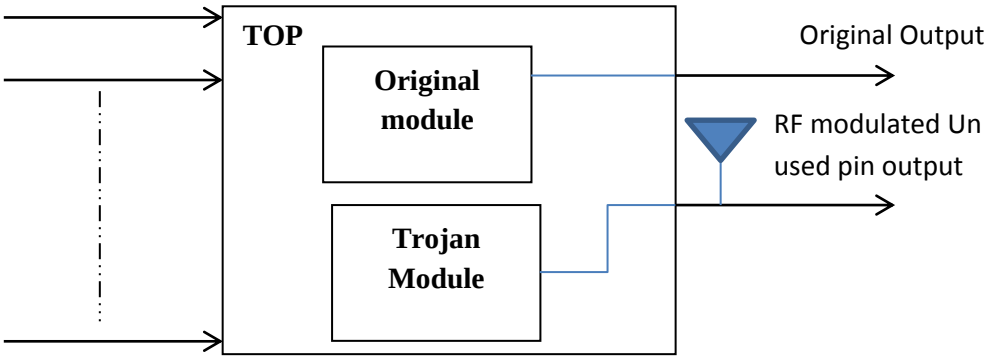

Figure 4-3 TSC using unused pin for RF transmission

Here also the Trojan triggering is based on a criterion. The criterion is encounter of either a particular state or sequence of states or after a particular number of encryptions. The Trojan after its trigger, transmits the amplitude modulated key through RF modulated unused pin as depicted in the figure.

DoS (Denial of Service):

Hardware attacks that force to DoS operations implies that the operation of the device is affected such that the hardware Trojan either forces the device to function incorrectly or not at all to function. Such kinds of attacks are very serious and needs to be taken care. This is because in some critical applications in medicine like pacemaker in which battery life time should be long, this DoS Trojan will drain the battery soon and leads to the death of the patient.

Leakage Current based TSC:

Trojan leaks the secret key through leakage current. The leakage circuit consists of a shift register and two inverters. Shift register holds the secret key. The LSB is connected to one inverter which is an input to the other inverter. When LSB of the shift register is '0', a path between power and ground is created by the PMOS of the first inverter and NMOS of the second inverter for a limited time. This increases the leakage current.

The attacker can easily determine the bits of key by measuring leakage current. The leakage circuit is common to all three benchmarks. Here the triggering criterion is either on the encounter of a particular state or particular sequence or after a particular number of encryptions.

4.2. POWER AND AREA ANALYSIS:

Twenty one AES benchmarks are analyzed for power and area using SYNOPSYS Design Compiler and are tabulated in the table. From the table it is evident that the area of Trojan free benchmark and the highest recorded area differ by **4941.348 square microns.**

Table 4-2 Dynamic Power and Area of AES Benchmarks

Benchmark	Dynamic Power	Power Delta	Area	Area Delta
AES-T100	185.0515	0.5311	379917.728	636.828
AES-T200	185.1306	0.6102	379936.448	655.548
AES-T300	184.5204	0	379280.888	-0.01199
AES-T400	184.7712	0.2508	380094.488	813.588
AES-T500	184.5204	0	379280.888	-0.01199
AES-T600	184.5315	0.0111	379408.328	127.428
AES-T700	187.481	2.9606	384222.2482	**4941.348**
AES-T800	187.3451	2.8247	384113.1681	4832.268
AES-T900	185.004	0.4836	381290.048	2009.148
AES-T1000	187.3202	2.7998	383795.2882	4514.388
AES-T1100	184.9933	0.4729	384096.9682	4816.068
AES-T1200	184.9933	0.4729	381311.648	2030.748
AES-T1300	184.5068	-0.0136	379397.168	116.268
AES-T1400	186.8748	2.3544	383387.0482	4106.148
AES-T1500	184.5225	0.0021	380389.328	1108.428
AES-T1600	187.6936	**3.1732**	384101.1282	4820.228
AES-T1700	184.6342	0.1138	382709.528	3428.628
AES-T1800	186.8379	2.3175	379280.888	-0.01199
AES-T1900	184.5204	0	379280.888	-0.01199
AES-T2000	184.5574	0.037	379694.888	413.988
AES-T2100	184.5159	-0.0045	380477.168	1196.268

Trojan Free Benchmark area: 379280.9 square microns

Trojan Free Benchmark Dynamic Power:184.5204 milliwatt

Power in milliwatt; Area in square microns; Delta: numerical difference between golden and trojan infected benchmark circuit

The Dynamic Power of Trojan Free benchmarks and the highest recorded power differ by **3.1732 mill-watt.** In the present sub-micron technology with billions of gates and transistors the recorded difference is not that significant to decide the existence of HT. Also small changes in the design constraints will have effect on the area. Hence Area and Dynamic power analysis don't serve the purpose of Hardware Trojan detection.

Procedure and Flow of Power Analysis:

If a design is to banalyzeded for net switching power, cell internal power, and leakage power Synopsys Power Compileare useded. Using Power compiler, Power Analysis can be done at two levels of abstraction.

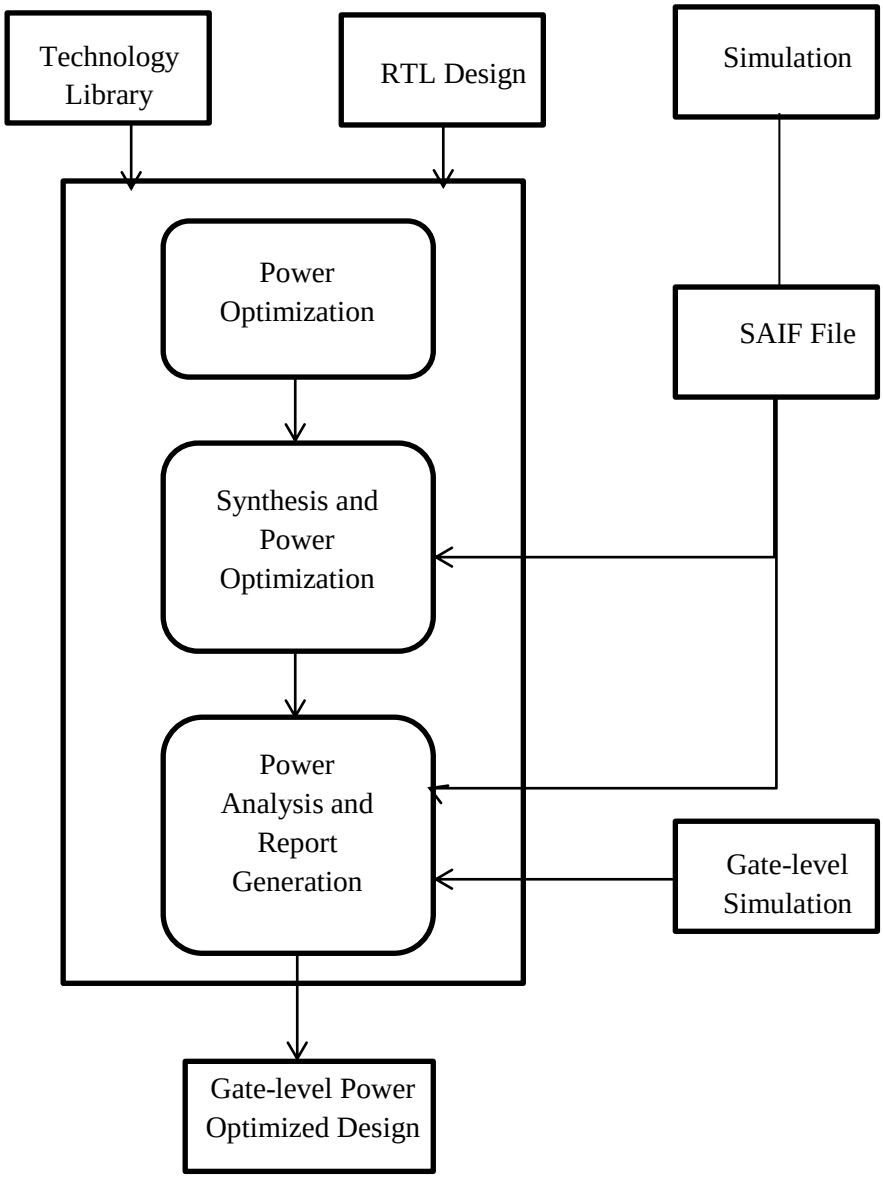

Figure 4-4 Power Analysis Flow

Switching activity from RTL or gate-level simulation is used for Gate-level Power Analysis. Analysis of a gate-level design requires a gate-level netlist and switching activity for it. Using Power Compiler one can determine the switching activity during RTL simulation. After captured activity on the design elements is annotated, switching activity is

propagated through un-annotated portions of the design. Switching activity from RTL simulation is much faster than that of gate-level simulation. But the power taken from the RTL simulation is not that accurate. For much accuracy Power from gate-level simulation is apt. For more specific power analysis a tool called Prime Power from Synopsys is used.

In general all the steps are the part of Synopsys Design Compiler but power analysis requires generation of a SAIF file that is generated during simulation. In Power Optimization layer techniques such as operand isolation and clock gating are used. Power Optimization plays a prominent role in sub-micron technology where billions of transistors and gates were fabricated in a small area. Synthesis and Power Optimization phase is done in the Synopsys DC-shell by integration of Design Compiler with Power Compiler. Power at several corners for the gate-level design can be done and a detailed report will be generated.

4.3. STRENGTHS AND WEAKNESSES OF AES BENCHMARKS:

Hardware Trojans based on CDMA:

AES-T100, AES-T200, AES-T700, AES-T800, AES-T900 AES-T1000, AES-T1100 and AES-T1200 fall in this category. In AES-T100 and AES-T200 the Trojan is always on. In AES-T700 and AES-T1000 the Trojan gets triggered after the occurrence of a particular state. In AES-T800 and AES-T1100 the Trojan gets triggered after the occurrence of a particular sequence of states. IN AES-T900 and AES-T1200 the Trojan gets triggered after 2^{128} number of encryptions.

This category of Trojans can't be detected in RTL verification as the functionality of the design is not at all affected due to the Trojan Side Channel. However, during DFT insertion, violations reported to asynchronous block will help suspecting the design. Upon properly identifying the suspicious blocks and removing them will pass the functional

coverage. Passing functional coverage will ensure us that the suspected block is a redundant malicious block.

All the asynchronous blocks reported are made synchronous for DFT (Design For Testability) insertion and pattern generation for stuck-at-faults and path-delay faults using ATPG (Automatic test Pattern Generator) of tetraMAX tool. The patterns generated by the ATPG are applied to the design that triggers the Trojan-trigger pin of the design by which the purpose of detection is served (Except AES-T100 and AES-T200).

Due to the asynchronous design of Trojan Side Channel block this class of Hardware Trojans even the bypass RTL verification gets detected during DFT insertion.

Hardware Trojans that perform DoS:

AES-T500, AES-T1800 and AES-T1900 are of this kind. They use Shift register rotation logic to drain the power of the device. The Trojan when triggered will continuously shift the shift register outputs circularly so that the battery will drain fast. Here the triggering criterion is either on the encounter of a particular state or particular sequence or after a particular number of encryptions.

These benchmarks, however as they do not interfere with the functionality of AES, it passes RTL verification. But when these benchmarks are synthesized for gate-level netlist a blank module with no inputs and outputs will be created. If code coverage is exercised on the synthesized gate-level netlists, for whatever the input combinations may be the particular module with no inputs and outputs will go uncovered. Re-checking the module with no inputs and outputs will direct to a shift register that is rotating the key unnecessarily to drain the power.

These benchmarks may be strong enough to pass the RTL verification, but are too weak as they get detected by synthesis followed by code coverage.

Hardware Trojans based on Leakage current:

AES-T600, AES-T2000 and AES-T2100 fall in this category. These Benchmarks based on a Triggering criterion leaks the secret key through leakage current. A Shift Register followed by inverter stages does the trick. The attacker can easily determine the key by quantifying the leakage current.

The leakage circuit for all the three benchmarks is identical. Since the side channel doesn't merge with the operation of AES it goes undetected in RTL verification. Like a DoS kind of Trojans the leakage circuit here doesn't have outputs. Hence Synthesis of the RTL design fir gate-level netlist will leave a Verilog module with no inputs and outputs which ultimately gets detected in code coverage like DoS. A close observation of the uncovered module will lead us to the Leakage Circuit.

Like DoS kind these benchmarks are strong enough to pass RTL verification, but the Trojan gets detected by synthesis followed by code coverage.

Hardware Trojans based on RF signal transmission:

AES-T400, AES-T1600 and AES-T1700 fall in this category. The data can be received at 1560 KHz with an AM radio. A single beep followed by a pause represents a '0' and a double beep followed by a pause represents '1'.

Unused pin in a chip is always to be suspected. In this category also the Trojan Side channel operation doesn't disturb original AES and performs its malicious operation concurrently. It is observed that the toggle coverage of all pins is not up to the level of

satisfaction until the Trojan gets triggered. As soon as the Trojan gets triggered it gets easily detected in RTL verification that includes code coverage followed by functional coverage.

Hardware Trojans based on the Dynamic Power Side channel:

AES-T1300, AES-T1400 and AES-T1500 fall in this category. The Trojan is designed such that it disturbs the key scheduling. The adversary intentionally introduces leaking states in the key schedule that depends upon known input bits and key bits. This does not occur during normal process.

The malicious behaviour in this category is that it leaks the key for every key schedule. The circuit that leaks the key is a shift register with ones and zeroes loaded alternately. Shift register gets enabled when the input to leakage circuit is logic '1', which results in additional consumption of dynamic power. The attacker decodes the power value to get the key.

This class of Trojans are strong enough to pass RTL verification as they do not disturb original AES operation. The lack of output port to the shift register leakage circuit will make the too weak to get detected in the synthesis phase itself. Similar to DoS and leakage current Trojans, this class of Trojans gets detected at the synthesis phase followed by code coverage.

The Powerful benchmarks:

Here comes the category of powerful Trojans. The word "powerful" is used here in the context that the malicious behaviour of AES-T100 and AES-T200 benchmarks is too difficult to get detected using straight forward verification techniques.

AES-T100 and AES-T200 passes all the standard verification process as their side channel is perfectly designed to bypass all the verification techniques.

Table 4-3 Summary of Detection of AES Trojans

AES-T-B	side-channel	Detection Details	Reason
AES-T100	power	Undetected	Passes all standard verification tests
AES-T200	power	Undetected	Passes all standard verification tests
AES-T300	power	Detected at synthesis stage	Lack of output pin
AES-T400	RF-Radio signal	Detected for toggle coverage at RTL verification	extra pin suspicious
AES-T500	power-draining battery	Detected at synthesis stage	Lack of output pin
AES-T600	leakage current	Detected at synthesis stage	Lack of output pin
AES-T700	power	Detected by patterns generated for SAF and PDF	Async Block, functionally passing
AES-T800	power	Detected by patterns generated for SAF and PDF	Async Block, functionally passing
AES-T900	power	Detected by patterns generated for SAF and PDF	Async Block, functionally passing
AES-T1000	power	Detected by patterns generated for SAF and PDF	Async Block, functionally passing
AES-T1100	power	Detected by patterns generated for SAF and PDF	Async Block, functionally passing
AES-T1200	power	Detected by patterns generated for SAF and PDF	Async Block, functionally passing
AES-T1300	power, leakage ckt, XOR	Detected at synthesis stage	Lack of output pin
AES-T1400	power, leakage ckt, XOR	Detected at synthesis stage	Lack of output pin
AES-T1500	power	Detected at synthesis stage	Lack of output pin
AES-T1600	RF-Radio signal	Detected for toggle coverage at RTL verification	extra pin suspicious
AES-T1700	RF-Radio signal	Detected for toggle coverage at RTL verification	extra pin suspicious
AES-T1800	power-draining battery	Detected at synthesis stage	Lack of output pin
AES-T1900	power-draining battery	Detected at synthesis stage	Lack of output pin
AES-T2000	leakage current	Detected at synthesis stage	Lack of output pin
AES-T2100	leakage current	Detected at synthesis stage	Lack of output pin

SAF: Stuck-at-Faults; PDF: Path Delay Faults

The summary of all the AES benchmarks and their detection status are tabulated in the table. The table clearly depicts all the benchmarks with their detection details and reason for their detection. From all the analysis done on the twenty one AES benchmarks taken from trust-hub, it can be concluded that Except AES-T100 and AES-T200 all other benchmarks get detected for standard verification techniques in ASIC. AES-T100 and AES-T200 go undetected at all stages of verification flow because the Trojans are designed in synchronous with the original design, doesn't affect the functionality of the original design and also possesses output port that avoids it get detected at synthesis level.

4.4. RSA TROJAN BENCHMARKS:

Three RSA Trojan Benchmarks with various malicious behaviours are taken from trust-hub for analysis and detection of Trojans inserted in them using existing conventional Hardware Trojan detection techniques. The description of three RSA benchmarks is tabulated in the table 4-4

Table 4-4 RSA Trojan benchmarks description

Benchmark	Malicious behaviour	Description
RSA-T100	Leaks Info	Whenever the input plain text is 32'h44444444 the output cypher text is replaced by secret key.
RSA-T200	DoS	Whenever a predefined input plain text occurs the in-exponent value is replaced with dummy value to perform DoS
RSA-T300	Leaks Info	After certain count value the Trojan triggers and replaces the cypher with secret in-exponent key
RSA-T400	Leaks info and DoS	After certain count value the Trojan replaces the secret key with a dummy value and leaks secret in-exponent key

DoS: Denial of Service

Based on the action characteristics the benchmarks possess malicious behaviour is that they either leak info of perform denial of service operation. RSA-T100 and RSA-T300 fall into the category of leak info and RSA-T200 and RSA-T400 falls in the category of DoS. A detailed description of these hardware Trojans is presented in the later section in this thesis.

4.5. STRENGTHS AND WEAKNESSES OF RSA BENCHMARKS

RSA Benchmarks that leak info:

The Trojans under this category leaks confidential info to the adversary. In this case the confidential info is either private key or in-exponent.

For the case of RSA-T100, the benchmark is designed such that the Trojan leaks the private key after a predefined input plain text is encountered. For input plain text of 32'h44444444 the Trojan will replace the output cypher text with the secret private key and serves the purpose of the adversary.

The behaviour of the RSA-T300 Trojan benchmark is similar to RSA-T100 but differs slightly in its activation mechanism. In this benchmark design, the adversary has incorporated a counter and the counter is initialized to zero. The counter count value is sensitive to the reset and data select pin. As soon as the counter's count value reaches to two, the benchmark enables its malicious behaviour by replacing the output cypher text with the in exponent to perform Denial of Service operation.

The corresponding simulation waveforms are shown in the figures.

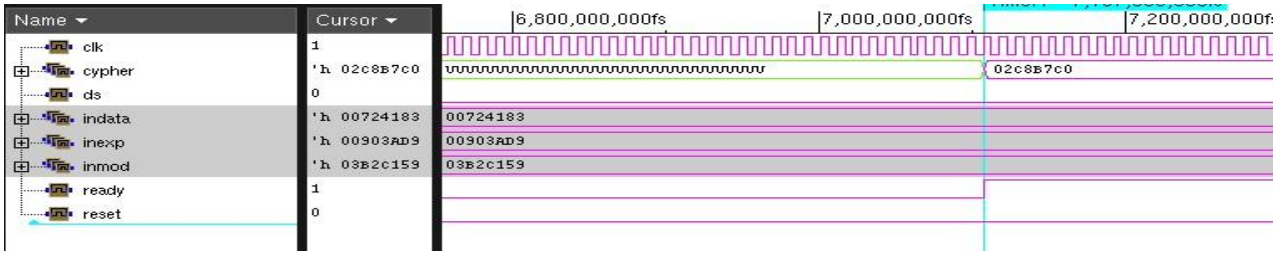

Figure 4-5 Simulation waveforms of RSA-T100 for no Trojan

The simulation diagram in the fig is for un-triggered Trojan. The Trojan is in-active as the input is 32'h00724183, a hexadecimal plain text for which the Trojan doesn't trigger. Hence the cypher text generated is flaw-less. It is known that the Trojan is active for input plain text of 32'h444444444 hexadecimal plain text.

Trojan Detection in RSA-T100:

The Trojan inserted in RSA-T100 can be detected in RTL verification itself because it is directly affecting the functionality of RSA module.

```
elsif count = 0 then
        if bothrdy = '1' and multgo = '0' then
                if indata ="01000100010001000100010001000100" then
                        cypher <= inExp;                -- Trojan leaks
                else
                        cypher <= tempout;              -- set output v
                end if;
                done <= '1';
```

Figure 4-6 Coverage report for RSA-T100

And hence for whatever may the input combinations achieving maximum coverage is impossible as at any instance at least if-condition fails and corresponding block coverage, condition coverage, expression and statement coverage will be affected.

Also the Trojan gets detected for the patterns generated by ATPG tool for Stuck-at-fault. The corresponding simulation waveforms are shown in figure.

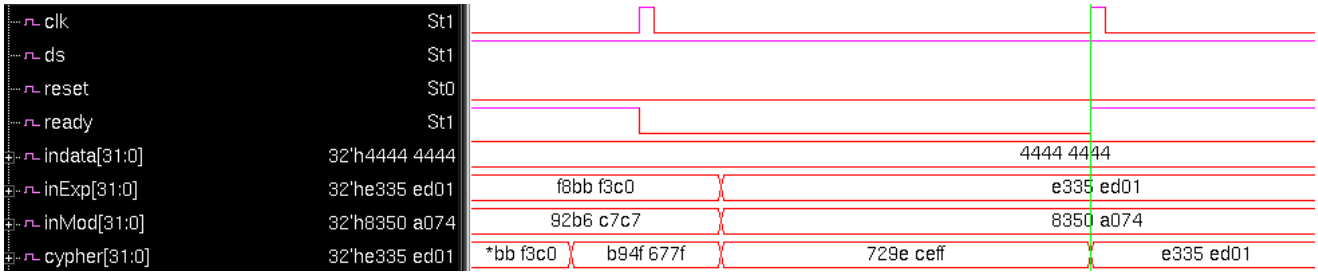

Figure 4-7 RSA-T100 Trojan triggered for pattern generation for SAF

4.6. *Trojan Detection in RSA-T300:*

In this benchmark the Trojan gets triggered after a particular count sequence value is brought up to two. Corresponding simulation waveforms of the benchmark are depicted in the figure for triggered Trojan.

Figure 4-8 RSA-T300 simulation waveforms for Trojan triggered

In the figure it is evident that, after some duration of time the cypher output is replaced with in-exponent. However this goes undetected at synthesis stage and pattern generation for SAF and PDF.

The Trojan inserted directly affects the functionality of the original RSA design and hence it gets detected for RTL verification. Standard verification metrics can easily detect this kind of Trojans.

Hence RSA-T100 and RSA-T300 benchmarks get detected for RTL verification. Thereby it is concluded that these benchmarks are weak.

RSA Benchmarks that perform DoS operation:

RSA-T200 falls in this category. The Trojan in this benchmark intentionally disturbs the original operation to perform denial of service operation.

The malicious block in this benchmark is designed such that on occurrence of a predefined input plain text the in-exponent value is replaced with a dummy value which is used in various evaluations required for cypher text calculations. The simulation waveform with Trojan triggered is shown in figure.

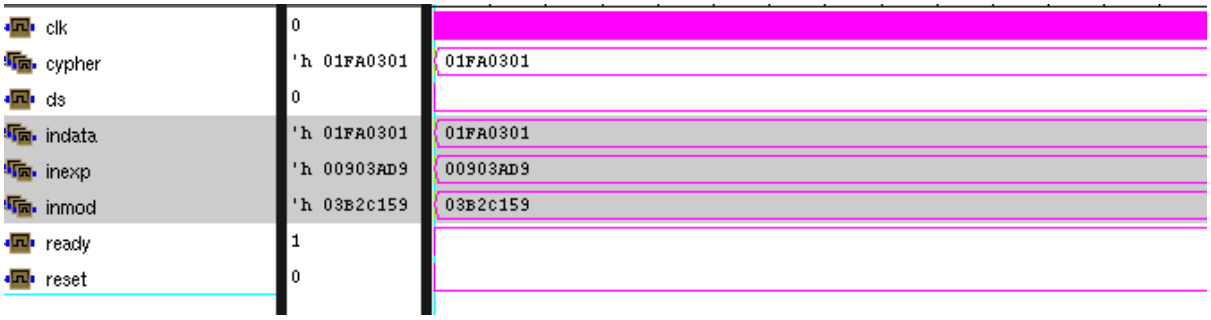

Figure 4-9 Simulation waveforms of RSA-T200 with active Trojan

From the input 32'01FA0301, the Trojan gets triggered and replaces the in-exponent value that in-turn alters the output cypher text.

Trojan Detection in RSA-T200:

The Trojan in this benchmark also alters the functionality of the original RSA and hence basic RTL verification can serve the purpose of detection. However this goes undetected at synthesis level and pattern generation for SAF and PDF as the Trojan is in synchronous with the original design.

In this benchmark the Trojan triggers for a predefined input and hence simple code coverage can easily detect the Trojan existence.

```
if indata = "00000001111111010000000011000000001" then      -- if
        inputExponent <= "00000000000000000000000000000001";
end if;
Trojan;
```

Figure 4-4-40 Coverage report for RSA-T200

The Coverage report clearly depicts that for whatever may be the input combinations, a condition in the Trojan module will be left uncovered as it is dependent on occurrence of a predefined input.

RSA Benchmarks that perform Leak info and DoS operation:

RSA-T400 falls in this category. When Trojan is triggered it replaces the in-exponent to perform the DoS and leaks the original secret in-exponent value. The corresponding simulation waveforms for active Trojan is shown in the figure.

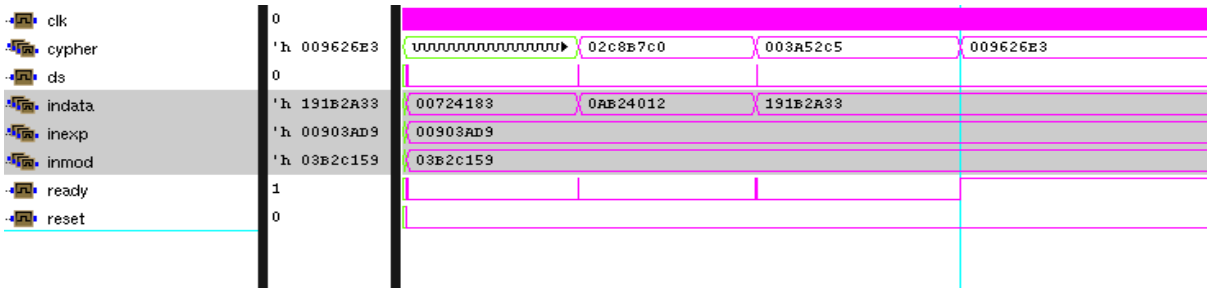

Figure 4-11 Simulation waveforms of RSA-T400 with active Trojan

From the figure it is evident that after the count value is brought up to two, the Trojan starts its malicious task of replacing the cypher with in-exponent and the cypher text will be replaced with original in-exponent value so that it performs both leakage of secret information and DoS operation as well.

Trojan Detection in RSA-T200:

This kind of Trojans bypasses detection at synthesis stage as the Trojan structure is perfectly organised to be synchronous with clock signal. Also it goes undetected for pattern generation for SAF and PDF.

However the issue with this benchmark is that it gets traced at RTL verification as the Trojan directly intervenes with the functionality of the original RSA benchmark. A perfectly written assertion based control oriented data coverage or data oriented coverage would serve the purpose of Trojan detection in this case.

On a whole all the RSA benchmarks taken form the trust-hub are not that powerful Trojans as they get detected at standard verification metrics like code coverage and functional coverage. Overall summary of these benchmarks is tabulated in the table.

Table 4-5 RSA Trojan benchmarks overall detection summary

Benchmark	Detection Status		
	Synthesis	SAF/PDF	RTL verification
RSA-T100	NO	YES	YES
RSA-T200	NO	NO	YES
RSA-T300	NO	NO	YES
RSA-T400	NO	NO	YES

CHAPTER 5

5. DESIGN OF NOVEL AES TROJAN BENCHMARKS

5.1. EXISTING DIFFICULTY

5.2. PROPOSED NOVEL BENCHMARK DESIGN

5.1. EXISTING DIFFICULTY:

From all the analysis done so far it is clear that almost all AES benchmark designs are weak enough to get detected in any of the standard verification technique. Hence there is a necessity for developing a novel hardware Trojan benchmark that bypasses all the Standard Trojan Verification techniques.

To achieve this task of designing a novel benchmark which is strong enough to pass the verification tests, a weakest Trojan class of all among the twenty-one benchmarks is selected. The weakest classes here imply that they even get detected at synthesis level which makes them too weak designs.

Hence Trojans based on Denial of service DoS, leakage current and dynamic power are taken and are re-designed to eliminate the weaknesses associated with them. The weakness associated with these benchmarks is that lack of output pin which lead to a blank Verilog module with no input and output gets created during synthesis that on further subjected to code coverage will do the job of Trojan detection.

To overcome the detection the design is to be altered such that output port is to be added to the design. Addition of output port is not as simple as it appears to be because the output port added to the TSC module should not affect the original output of the TOP (AES and TSC) module.

Before that a detailed study of the DoS based Trojan benchmark is presented in the later sections. To illustrate the idea, AES-T500 benchmark is taken as reference. The simulation result of AES-T500 trojan inserted benchmark, for both Triggered and un-triggered states, is shown in the figure.

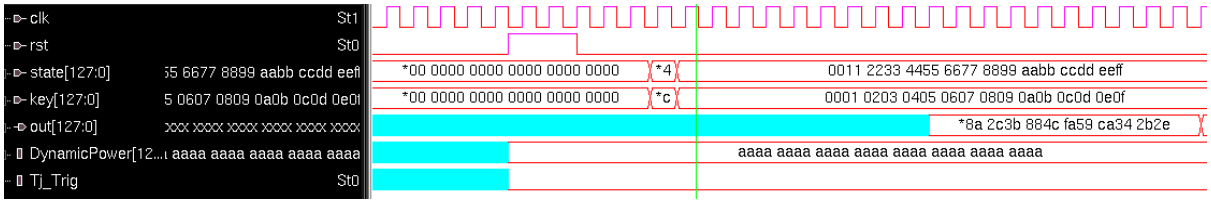

Figure 5-5-1 Simulation waveforms of AES-T500 for un-triggered Trojan

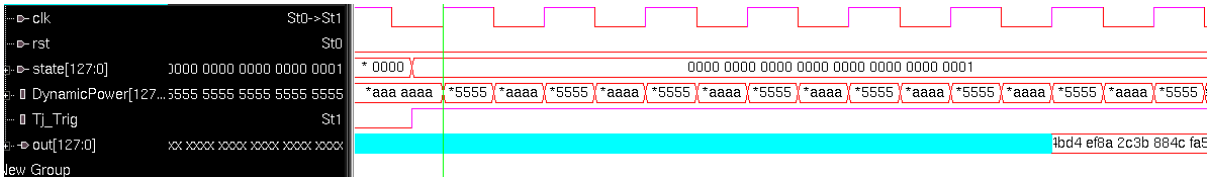

Figure 5-5-2 Simulation waveforms of AES-T500 for Triggered Trojan

The simulation wave for shown in figure corresponds to that for un-triggered Trojan. The pin *Tj_Trig* is the one that actually depicts the status of the Trojan that whether the Trojan is in triggered state or un-triggered state. For this particular benchmark design, the context for which the Trojan triggers is occurrence of a predefined sequence of 128-bit plain text at the state input. If once the sequence not as pre-designed then the Trojan remains un-triggered that is shown in figure.

Trojan Structure:

The Trojan structure is that, it is a 128-bit shift-register. The shift-register structure here is not at all a part of AES design. It is a redundant block. The shift-register is initialized with an appropriate output value of 128-bit. In this case it is 128'haaaa_aaaa_aaaa_aaaa_aaaa_aaaa_aaaa_aaaa. The malicious action of the shift-register circuit is that up on the commencement of predefined sequence of states at the input of the AES, the *Tj_Trig* pin immediately goes high and the shift-register starts to shift the data circularly. The malicious operation once started doesn't get terminated until unless the condition for the *Tj_Trig* pin to go low occurs.

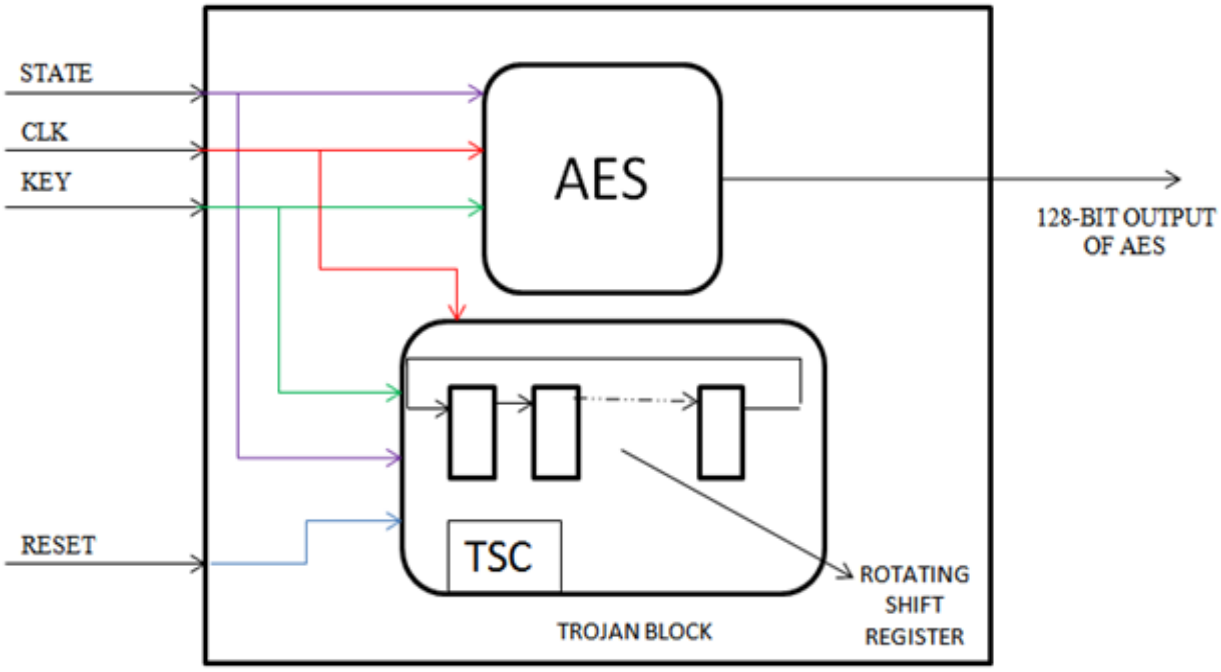

Figure 5-5-3 The structure of Trojan in AES-T500 benchmark

The design of Trojan block in AES-T500 is shown in figure. In addition with AES benchmark the malicious block that is accessing the state input, clock input, key input and reset input. The rotating shift-register is designed in ring counter style.

This class of Trojans are very weak as they can be detected at synthesis level of abstraction itself. This is because of lack of output pin for the TSC module. A simple solution that hits is addition of output pin to design would serve the purpose of bypassing the synthesis phase. But the problem here is addition of additional pin should not affect the output of the top module. This is because if the added extra port disturbs the original output, the Trojan will be weak enough to get identified at RTL verification stage. Hence the addition should be designed in such a way that any result of the outcome that results in altering the original output should be nullified.

5.2. PROPOSED NOVEL BENCHMARK DESIGN:

A Novel AES benchmark design is proposed for the classes of DoS, Leakage Current and Dynamic Power based Trojan designs so that they bypasses all standard verification methods of Trojan detection. Especially the synthesis phase is to be bypassed as they get caught at this level of abstraction due to lack of output pin.

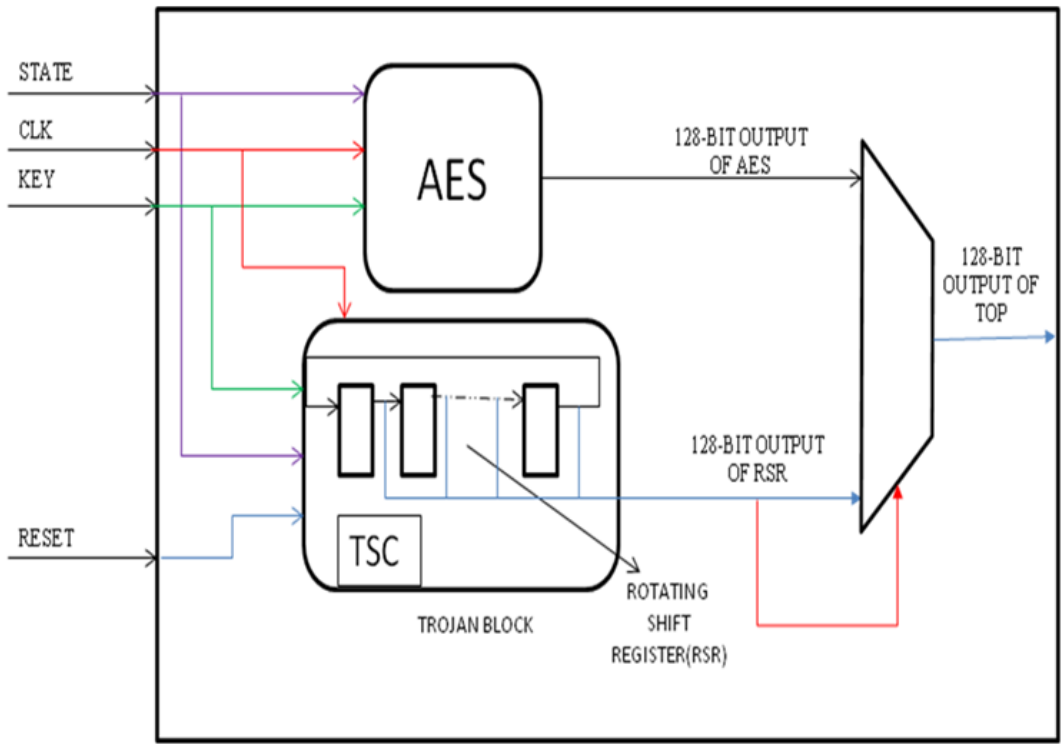

Figure 5-4 Proposed structure of Trojan Benchmark design for DoS, Leakage Current and Dynamic Power classes.

The proposed design is actually the modification of the existing design incorporating necessary changes. The key idea here is to add an output pin to the design and incorporate a multiplexer whose inputs are the outputs of AES module and TSC module's newly added output. The output pin is of size 128-bit and the output is actually the output of the Rotating Shift Register (RSR).

The select line used to the multiplexer is the newly added output of the TSC module. The multiplexer logic here is to select the output of AES for all values of select line except for zero. The select line which is output of TSC module will never go zero and hence the purpose of un-altered original output value is satisfied.

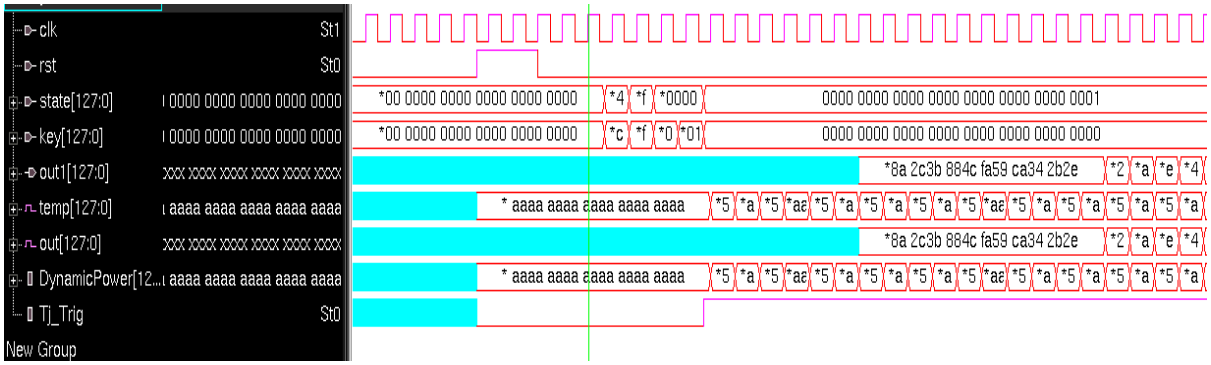

Figure 5-5 Simulation waveforms of proposed novel design

The simulation waveforms of the proposed design is show in figure from the figure it is evident that the output value doesn't change due to added output to the TSC module due to multiplexer with one of the inputs and select line as the output of the Rotating Shift Register. This new benchmark design doesn't get detected at synthesis level die to added output port and also standard verification techniques like code coverage and functional coverage cannot detect the Trojan in them as the AES inside it is functionally unaltered.

CHAPTER 6

6. CONCLUSION AND SCOPE OF FUTURE WORK:

6.1. CONCLUSION

6.2. FUTURE WORK

6.1. CONCLUSION*:*

After a detailed post-mortem of the twenty one AES Trojan benchmarks taken from the repository of trust-hub, it is very clear that the Trojan designs are not that effective to bypass the standard verification techniques. As a matter of fact except Always on Trojans all the other class of Trojan designs gets caught somehow at any level of abstraction easily.

All the Trojan benchmarks were subjected to DFT insertion and pattern generation using ATPG, RTL verification for all the benchmarks was done. As no benchmark is directly affecting the functionality of the original AES design it is very difficult to write a powerful assertion based test benches. Hence these Trojan benchmarks bypass RTL verification. In case of RSA benchmarks as the Trojan designs are directly intervening in the functional operation, the gets easily detected for RTL verification.

Table 6-1 Trojan Category and its strength

Serial no.	**Side Channel Category**	**Strength**
1	CDMA Based	WEAK
2	RF Transmission based	WEAK
3	Leakage Current Based	WEAK
4	DoS Based	WEAK
5	Always on Category	STRONG

Each and every AES Trojan benchmark is associated with certain weakness that leads to their detection. From all the work carried out the benchmarks are too weak enough to get caught in case of ASIC. However for a good Trojan designer, his malicious blocks should not

get caught elsewhere. A fruitful attempt was made in order to overcome the weaknesses of AES benchmarks out of which the categories of Dynamic Power, leakage current and DoS were successfully redesigned and the designs were subjected to existing standard verification metrics and got assurance that the designs will never get caught for the standard verification metrics known to the best of my knowledge. Due to lack of time the work is not extended for the other categories like CDMA based Trojans and RF transmission based Trojans.

6.2. FUTURE WORK:

The work done so far is with respect to Application Specific Integrated Circuit ASIC design flow. The work can be extended to verify the strengths and weaknesses of AES benchmarks and RSA benchmarks with respect to Field Programmable Gate Array (FPGA). Also various Trojan defense architectures can be incorporated in these benchmarks to thwart the malicious behaviour.

Bibilography:

[1] "Report of the Defense Science Board Task Force on High Performance Microchip Supply", Defense Science Board, US DoD, Feb.2005; http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf.

[2] M.Tehranipoor, Farinaz Koushanfar, A survey of Hardware Trojan Taxonomy and Detection, IEEE Design and Test of Computers-2010.

[3] S. Adee, "The Hunt for the Kill Switch", IEEE Spectrum, vol.45, no.5, 2008, pp.34-39.

[4] Stefan Mangard, Elisabeth Oswald, Thomas Popp, "Power Analysis attacks-revealing secrets of smart cards", Springer Science-2007.

[5] B. Yang, K.Wu, R.Karri, Secure Scan: A Design for Test Architecture for Cryptochips, Design Automation Conference (DAC 2005), Anaheim, July 12-14 pp135-140, 2005.

[6] Y.Jin and Y.Markis "Hardware trojan detection using path delay fingerprint", IEEE intl. workshop on Hardware oriented security and trust, pp.51-57,2008.

[7] Swarup Bhunia et.al, "Protection against Hardware Trojan Attacks: Towards a comprehensive solution", IEEE Design and Test-2012.

[8] Adam Waksman, et.al, "Practical, Lightweight Secure inclusion of third party Intellectual Property", IEEE Design and Test-2013..

[9] M.Bilzor et.al, "Evaluating Security requirements in a general purpose processor by combining assertion checkers with code coverage" IEEE International Symposium on HOST, San Francisco, CA, June 2012, Pages 49-54.

[10] R.S.Charkraborty et.al, "MERO: A Statistical Approach for hardware trojan detection", Proc. 11th Intl. CHES Workshop-2009

[11] Seetharaman Narasimahan, et.al, Hardware Trojan Detection by Multiple Parameter Side-Channel Analysis, IEEE Transactions on computers, vol.62, No.11, November-2013.

[12] https://www.trust-hub.org/resources

[13] Xuehui Zhang and Mohammad Tehranipoor, Case Study: Detecting Hardware Trojans in Third-Party Digital IP Cores, IEEE International Symposium on HOST-2011.

[14] Trey Reece, William H Robinson, "Analysis of data-leak hardware Trojans in AES cryptographic circuits" Technologies for Homeland Security (HST), 2013 IEEE International Conference on pp. 467-472

[15] M.Banga and M.Hsiao, "A novel sustained vector technique for the detection of hardware trojans" , Proc. 22nd Intl. conf. VLSI design, IEEE CS press, 2009, pp. 327-332.

[16] P.Rohatgi, "Improved techniques for side-channel analysis", cryptographic engineering, pp.381-406-2009.

[17] E. Love, Y. Jin and Y. Makris, "Enhancing Security via Provably Trustworthy Hardware Intellectual Property," Proceedings of the IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2011.

[18] L. Lin, M. Kasper, T. Güneysu, C. Paar and W. Burleson, "Trojan Side-Channels: Lightweight Hardware Trojans through Side- Channel Engineering," 11th International Workshop Cryptographic Hardware and Embedded Systems (CHES), pp. 382-395, 2009.

[19] Alex Baumgarten, Michael Steffen, Matthew Clausman, Joseph Zambreno, "A case study in hardware Trojan design and implementation," International Journal of Information Security, Volume 10, Issue 1, pp. 1-14, 2011.

[20] Synopsys user manuals for Design Vision, VCS and Tetramax

[21] Computer and Network Security by Avi Kak