2011

Electromagnetic Side-Channel Analysis for Hardware and Software Watermarking

Ashwin Lakshminarasimhan
University of Massachusetts Amherst

ELECTROMAGNETIC SIDE-CHANNEL ANALYSIS FOR HARDWARE AND SOFTWARE WATERMARKING

A Thesis Presented

by

ASHWIN LAKSHMINARASIMHAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial
fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

September 2011

Department of Electrical and Computer Engineering

ELECTROMAGNETIC SIDE-CHANNEL ANALYSIS FOR HARDWARE AND SOFTWARE WATERMARKING

A Thesis Presented

by

ASHWIN LAKSHMINARASIMHAN

Approved as to style and content by:

_____

Wayne P. Burleson, Chair

_____

Christof Paar, Member

_____

Kevin Fu, Member

_____

C.V. Hollot, Department Head
Department of Electrical and Computer Engineering

DEDICATION

To my family

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Professor Wayne Burleson for providing me with the opportunity to expand my knowledge of electrical and computer engineering, and for his constant guidance and support throughout my thesis research. I would also like to thank Professors Wayne Burleson, Christof Paar, and Kevin Fu for their service on my thesis committee. I also extend my gratitude to the entire VLSI Circuits and Systems group for both their friendship and academic support, especially Georg Becker, and Lang Lin who were involved in a number of discussions and provided help in times of difficulty. I also thank my friends and family for their constant support and encouragement.

ABSTRACT

ELECTROMAGNETIC SIDE-CHANNEL ANALYSIS FOR HARDWARE AND SOFTWARE WATERMARKING

SEPTEMBER 2011

ASHWIN LAKSHMINARASIMHAN

B.E., ANNA UNIVERSITY, CHENNAI, INDIA

M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Wayne Burleson

With more and more ICs being used in sectors requiring confidentiality and integrity like payment systems, military, finance and health, there is a lot of concern in the security and privacy of ICs. The widespread adoption of Intellectual Property (IP) based designs for modern systems like system on chips has reduced the time to market and saved a lot of money for many companies. But this has also opened the gates for problems like product piracy, IP theft and fraud. It is estimated that billions of dollars are lost annually to illegal manufacturing of Integrated Circuits. A possible solution to this problem of IP theft is to insert small circuits which are like unique IDs that only the owner or the registered verifier will know and detect in case of any conflict. The circuits that are inserted are called watermarks and are in some cases kept very small so as to be hidden. In such cases, we would need detection schemes that work well even with very small watermarks. In this work, we use Electro-Magnetic (EM) based side-channels for the detection of watermarks. Since the 90s, Side-channel Analyses have attracted significant attention within the cryptographic community as they are able to obtain secret information from smart cards and ICs. The power side-channel analysis is a very

powerful method but EM side-channels are very useful as they will not need a resistor in series to the power supply and just needs passive observation of the EM radiations emanated by the IC. This passive monitoring will be a big advantage in the case of automated watermark detection used by a verifier.

In this work, we start with EM side-channel analysis on FPGA for smaller designs. We insert watermarks on a Microcontroller, Smartcard and an FPGA and detect these watermarks using EM side-channel information emanated from the Design under Test. We used environments with different levels of noise interference. We compare the watermarking application using EM side-channels and Power side-channels in these different setups. These watermarks are very small and are hard to attack or remove by an attacker through reverse engineering or side-channel information. Due to the robustness against such attacks and the easy access of EM side-channels when compared to power side-channels, the EM side-channel based watermarks will be a very good solution for the IP theft problem. EM side-channel based watermark detection supports automation which companies of IP cores can make use of. We also extended this work to EM Side-channel Trojans as the concepts are similar.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The aspect of security to embedded systems is very important nowadays as more and more systems are being used in the fields which need secrecy and confidentiality, such as medicine, military and finance. The growing number of breaches in all the above fields presents unique challenges and requires utmost attention in order to prevent them. Security for embedded systems can sometimes be misinterpreted as only adding secure cryptographic algorithms which keep the key secret and mask the information in use. Though any embedded system would require that as a pre-requisite nowadays, there are more things that should be taken care of. The embedded systems should take into consideration issues like the possibility of attacks on the algorithms, attacks on the implementation methods, insider threats and property misuse.

1.1 Watermarking

The widespread adoption of Intellectual Property (IP) based designs for modern systems like system on chips have reduced the time to market and saved a lot of money for many companies. But this has also opened the gates for problems like product piracy, IP theft and fraud. This work focuses on the security aspects of an IP when a legitimate owner wants to protect his IP rights and make it exclusive by inserting a very small design which will be a unique ID or key for his IPs. This can be considered a nametag for the IP, which is generally kept hidden so that proving its existence can be possible by the owner or the verifier who has the rights to find the IP identity. This method is called

Watermarking and is applicable to both Hardware chips and Software codes and programs. To protect ASIC and FPGA chips, we would need watermarks which are called *Hardware watermarks*. We know that code theft is often possible in the case of PC software, and to avoid this from happening in embedded systems like microcontrollers and smartcards, we extend the above mentioned watermarking techniques to programs (software) used. In the case of smartcards, which are under attack more often than the other systems, the usage of *software watermarks* is needed. In the case of Software watermarks, the codes which are used for programming the smartcards are modified. The original code, for example the 3DES algorithm is modified to have a small portion of code that acts as the watermark. In order to prove one's ownership, one has to leak out this small portion of code which only the owner can know in detail.

Watermarks can then help the original party to prove that it is his design that is being used by someone else when times of conflict arise. It is to make sure that the owner of the IC has the exclusive rights and no one else can claim ownership of this IC. For example, let us say that company X owns a design D_x and it inserted specific watermarks W_x. When another company Y obtains the IP core D_x, it will not know that there are watermarks W_x contained in D_x. And if it involves in illegally claiming the D_x design as its own, Company X can then prove that Company Y has illegally used its design by detecting the watermarks W_x and proving it to the jury. This way one can protect his design from being used by another party without consent. There is also the problem of IP counterfeiting wherein a fraudulent party can sell its sub standard IPs under the name of a well established party. Watermarks do not prevent illegal copying or counterfeiting and hence this type of IP problem is not our focus in this work. The inserted watermarks can

however find if there is any abuse of IP once there is a doubt. These are hidden by the owner of the design so that he can prove his identity in times of conflict and IP fraud.

1.2 Usage of Side-channels in Watermarking

Most of the watermarks are very small so that they can be kept hidden within the big IP design of the ASIC in the case of hardware or program code in the case of software. In the case of hardware, we insert a small logic that makes use of a unique key, performs a combinational logic with this key, and makes this information available to the owner or verifier. In the case of software, we will insert a few lines of code that is based on the same principle of using a unique ID / key and a function along with it. The watermark design will be small when compared to the existing IP design. In such cases, the detection of such watermarks should be efficient, but can be tricky. In the case of media watermark like images and videos, signal processing methods are used. Similarly, for watermarks used in integrated circuits and program codes, there should be efficient methods that can detect the watermark clearly.

When an IC is computing an operation, it consumes power, generates heat, and also emits radiations. For example, when a processor is used for computation for sometime it tends to heat up. This is due to the components of the IC in use. Thousands of transistors in use obviously lead to consumption of power and generation of heat and radiations. When ICs that compute a crypto algorithm are considered, these radiations and power consumption do not represent random energy spent but can be dependent on the data or operation computed, and can be exploited to reveal the secret. For example, the logic value a transistor can pass can be a 0 or a 1. The power consumption and

radiation emitted will be different for 0 and 1, and this information can be correlated to reveal whether the key bit is a 0 or 1. Thus, if the radiations and power consumption are data dependent, it becomes a huge vulnerability. Paul Kocher made use of these vulnerabilities. He proposed differential power analysis [3] that worked based on the dependency of power consumption on device activity. These physical measurements can then be provoked and monitored by adversaries which results in important information which could then be exploited. This important information is termed *Side-Channel information* and attacks targeting such information are called *Side-Channel Attacks*.

This side-channel information can be used in the case of watermarking. As we know every design will create side-channel information which can be power consumption or EM radiations. So when we insert a new circuit, the side-channel information will vary because there is an extra logic that can create different power consumption or EM radiations. Every design will create side-channels and even a small watermark will have its own impact on the overall side-channel from the design. When we consider the side-channels, there is some difference between the side-channel information from design D_x and the design that has watermark W_x in it ($D_x + W_x$). This difference in the side-channel information can then be used in the detection of watermark that was inserted.

Detection of such side-channels can prove the existence of the watermark and help solve the problem of IP fraud. This thesis focuses on watermarking application and the use of EM side-channels for the detection of such watermarks. We will now discuss side-channel information and what analysis is required for the detection of watermarks.

1.3 Side-channels

Side-channel analysis is a new research area that has gained more and more interest since the late nineties. The research in this area has shown that physical leakage caused by the implementation of a secure mathematical algorithm can be crucial in terms of security. For example, such a leakage can be sufficient to extract secret key material from cryptographic implementations. Different Crypto algorithms may follow different methods and steps to encrypt and decrypt a message, but the heart of any algorithm remains the secret key. Intermediate values that depend on part of the keys are the preferred target of the attackers. Attacks are performed on this information when a cryptosystem is running its normal operation. In cryptanalysis, a side-channel attack is the class of attack where an attacker will try to deduce what is occurring inside a device by observing the information that leaks during the normal functioning of the device rather than brute force or theoretical weaknesses in the algorithms. That is to say the side-channel attacks do not prove anything about the security provided by the mathematical and technical functions involved, but rely on how well they are carried out without leakage. For example, timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information which can be exploited to break the system. Like the power side-channels, EM side-channel is an important information source and is available when any system operates. The sources of such EM radiations are given below [6], [8]:

- Direct Emanation

These emanations result from the intentional current flows within the circuits. The current flow takes place in the circuit when it performs its assigned function. This

intentional current flow generates time-varying electric and magnetic fields which are related by Maxwell's equations. In CMOS circuits, these current flows consist of short bursts of current with sharp rising edges that occur during the switching operation and result in EM emanations that can be observed over a wide frequency band.

- Unintentional Emanation

This kind of emanation arises from the fact that most modern devices pack a large number of circuits and components into a very small area and suffer from numerous unintentional electrical and electromagnetic couplings between components, depending on their proximity and geometry. Such couplings are a rich source of emanations that can be compromised and are being neglected by the majority of the designers since these couplings do not affect the functionality of the device. With technology scaling, these unintentional emanations are going to rise considerably and thus are going to be targeted by attackers.

The side-channel information from each device can be retrieved via 2 methods which are classified based on their approach as:

1) Active methods
2) Passive methods

The active method as the name indicates, involves getting information from the device by tampering with the outer layer of the device and then analyzing how the device behaves, extracting information by placing probes on top of the different parts of the device.

On the other hand, just observation of the radiations and leakages from the device without even tampering with the device can provide information that could be used and is called passive, as it does not involve any active tampering. This passive method is what we will deal in this work. There are different sources that can be passively used. They are listed below according to the year when they were first accomplished.

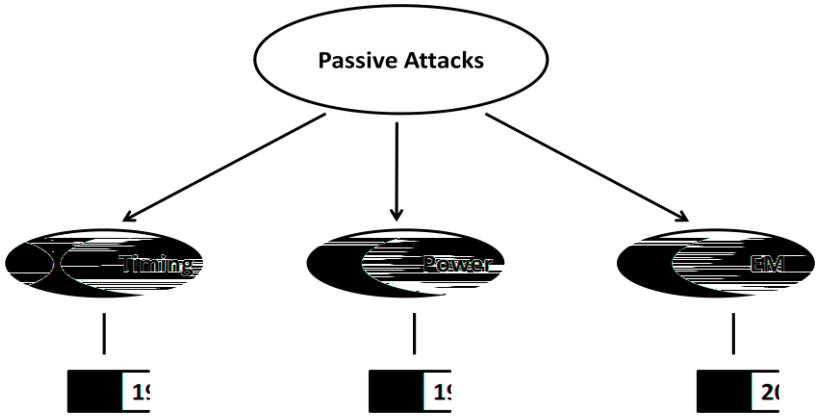
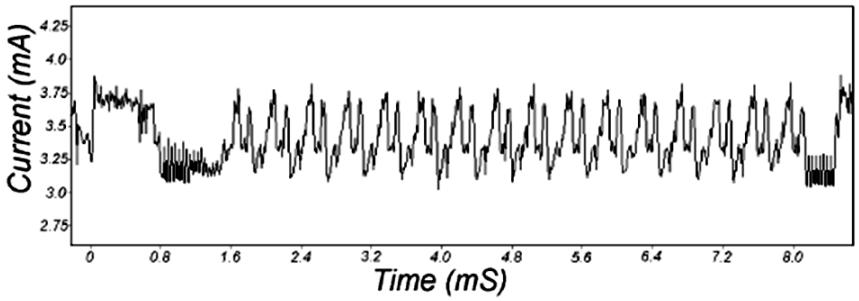


Figure 1.1 Types of passive attacks

Although these are extremely efficient, side-channel analyses require considerable technical knowledge of the internal operation of the system on which the cryptography is implemented. These side-channel analyses when classified *based on their complexity* are listed below,

1. Simple Side-channel Analysis
2. Differential Side-channel Analysis

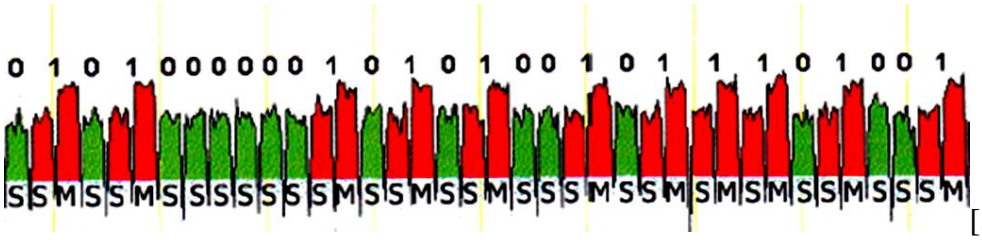
Simple Side-channel Analysis is that which makes use of the information that is directly available from a measurement trace. This side-channel analysis is useful when the power or EM side-channels from different operations are easily distinguishable by visual inspection. For example, in case of a micro controller performing addition and multiplication operations, the power/EM signal consumed when performing addition is different from the power/EM signal consumed when performing a multiplication and can be easily observed. If crypto algorithms like DES or AES are used, visual inspection should lead to first finding out where the different rounds are, although knowing the algorithm and where the rounds are is not considered as an attack.



[3]

Figure 1.2 Power signal showing 16 rounds of DES operation

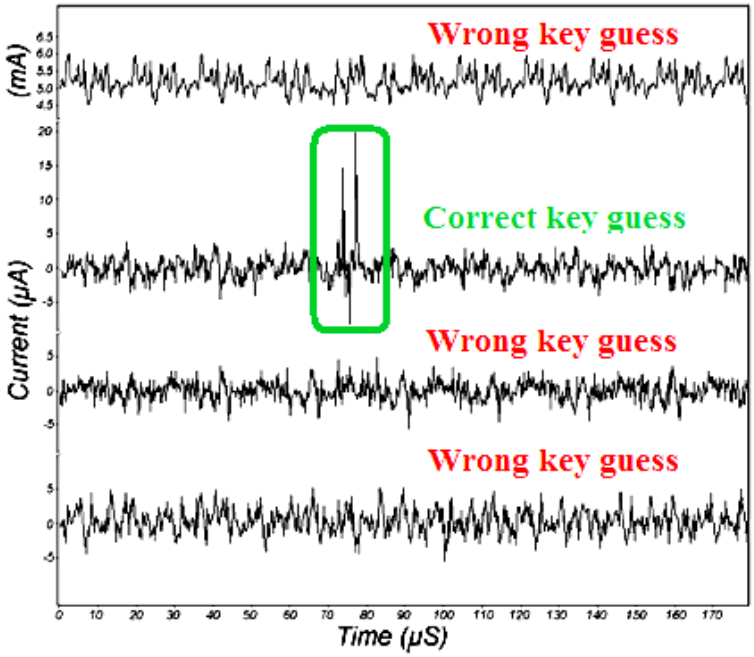
The implementation of operations that generate traces having key dependent branching can be analyzed with this method. Here the key has a simple relationship with the operations that are visible in the trace. The following diagrams show clearly how simple power analysis is achieved.



[31]

Figure 1.3 SPA of Square Multiplication operation

Differential Side-channel Analysis is the method where the side-channel differences are not directly visible in one measurement trace. This side-channel attack needs a number of measurement traces with different inputs. This requires statistical methods to be applied. One specific intermediate result such as an S-box is targeted. The analysis is continued with different guesses of the key until the correct key guess is obtained. The guesses are made and the corresponding traces are divided into 2 groups based on the guess for a bit to be a '0' or a '1'. All the traces in group 1 are added together and all the traces in group 0 are added together. Then they are averaged and the difference between the averages of group 1 and group 0 is calculated. If the key guess made is correct, we can expect to see a considerable amount of spike or peak and for other incorrect key guesses, the difference of averages would yield only random noise and not a peak. Figure 1.5 shows the correct and wrong key guesses.



[3]

Figure 1.4 DPA waveform with correct and wrong key guesses

Initially, these analyses methods were mainly focused on attacking an embedded system. There are also the concepts of Trojans inserted by an insider which aids leaking out the secret information without the concerned party's knowledge. The Trojans are small circuits that an insider inserts into the original clean design so that it will leak out the secret information to the outside world. These Trojans can be of different types and are explained in [46]. The watermarks inserted in our case follow concepts and designs similar to that of the Trojan, but for a constructive purpose.

1.4 Motivation

The side-channel information was initially used to attack systems but it could also be used in a constructive way in order to provide Intellectual Property rights and

privileges where the concept of Trojans is flipped around to help the owner. Usage of side-channel for constructive purposes is limited to power side-channels and mainly for micro controllers. Usage of EM side-channels for Hardware Trojans and Hardware and Software Watermarks has not been attempted before and provides the motivation for this thesis.

In this work, we propose to use watermarking using Electromagnetic (EM) side-channel, which is similar to the power side-channel but has an important advantage of not requiring inserting a resistor in the power supply path to monitor power consumption. Using EM side-channels could eliminate the requirement to actively probe the device. Instead placing a probe passively on top of the device in operation could lead us to the watermarks.

The main contribution from this work is the implementation and detection of very small watermarks in ubiquitous devices like microcontrollers and smartcards, as well as in FPGA. Also this work proves that EM side-channel can detect the watermark clearly and requires no clean setup and active access to the device. We validate this by providing results from both Power and EM side-channel watermark detection. EM side-channels if employed by the manufacturer, can check for the watermarks by using an automatic work flow involving mechanical arms for the devices, and need not make any active contact, such as the power side-channel based watermarking schemes. This can lead to faster validation of the devices from the manufacturer's perspective. Due to increasing use of different Intellectual Property (IP) cores from different companies in order to build a new Embedded system or ASIC, there is a need to make sure there is efficient security in place to avoid IP theft, counterfeiting and frauds. Digital watermarking for integrated

circuits has been proposed in the past as a solution to identify the owner and detect frauds, if any. We propose to use existing watermarking principles, and use EM side-channel information for its detection. These concepts are carried out using EM side-channels for the first time in this work and explained in the next chapters.

1.5 Thesis Outline

In this thesis we will begin with the history and prior work of Electromagnetic Side-channel Analysis and Watermarking. Chapter 3 explains the set up of an experimental platform for Microcontroller, Smartcard and FPGA along with the exploration of some processing techniques. Chapter 4 explains how EM analysis was carried out on a Spartan 3E FPGA. Chapter 5 discusses the usage of Hardware Trojans to attack algorithms. Chapter 6 explains the usage of EM side-channel with Watermarking for Microcontrollers, Smartcards and FPGA.

CHAPTER 2

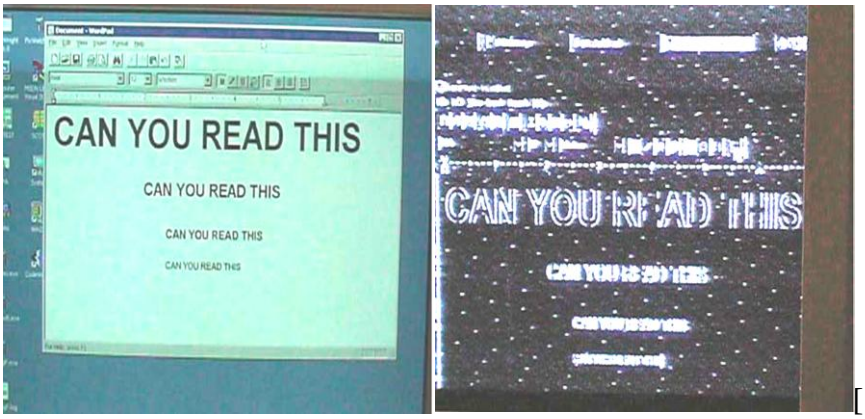
HISTORY AND PRIOR WORK

In this chapter, we will first discuss the history and prior work of side-channel based attacks and how side-channels could be used for watermarking applications. The first official information related to Side-channel usage dates back to the year 1965. P. Wright reported in his book "*Spycatcher: the candid autobiography of a senior intelligence officer*" that the British intelligence agency tried to break a cipher used by the Egyptian Embassy in London, but initially their efforts were hindered by the limits of their computational power. So in order to overcome the problem, Wright suggested placing a microphone near the rotor-cipher machine used by the Egyptians in order to spy the click-sound produced by the machine. They then listened to the clicks of the rotors every morning when the clerk reset them. Thus MI5 could successfully deduce the position of 2 to 3 rotors of the machine. This became additional information which considerably reduced the effort in terms of computation needed to break the cipher. This also led the MI5 to spy on the embassy's communications for many more years.

2.1 Side-channel Usage for Attacks

Even the EM side-channel has a history of rumors associated with espionage. The defense organizations are obsessed with limiting the electromagnetic emanations from the equipments they use. There are also regulations for commercial ICs about how much emissions can be allowed without interfering with the proper operation of the chip and are classified into the Electromagnetic Interference/ Electromagnetic Compatibility (EMI/EMC) areas. Every company places great importance to satisfy these conditions

while manufacturing chips. The importance of EM side-channel was first openly stated by Van Eck in 1985 [1] when he made use of the EM emanations from a computer or a video unit captured from a distance and used them to retrieve the information that was displayed on the screen.



[1]

Figure 2.1 EM signals captured from a computer and reconstructed

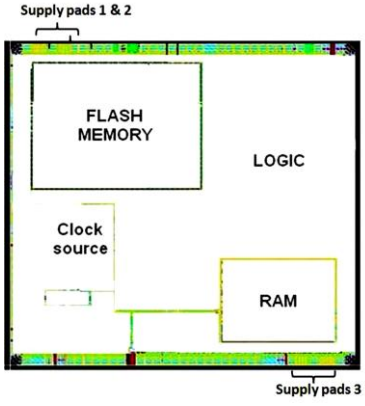
Figure 2.1 depicts how the display from a computer was reconstructed using EM. But the first side-channel attack on a crypto system was made in 1995 with the advent of the use of smart cards and microcontrollers for parking tickets, payment for transportation and transactions. In 1995, Timing analysis was the first side-channel based attack ever published. Paul Kocher [2] described the methodology to compromise keys of RSA, DSS and other cryptosystems by measuring the execution time for the overall cryptographic operation. This attack required an attacker to be able to simulate or predict the timing behavior of the attacked device rather accurately. In 1998, a much more efficient side-channel was introduced by Paul Kocher, Joshua Jaffe, and Benjamin Jun- Differential Power Analysis [3]. The power consumed by a cryptographic device was analyzed during the processing of the cryptographic operation. The power consumption

turned out to include deterministic data-dependent parts which can be exploited by Simple Power Analysis and Differential Power Analysis. From then on, specific parts of the implementation of the cryptographic algorithms are being observed. For instance, Simple Power Analysis looks at the key scheduling process; Differential Power Analysis chooses an intermediate result for testing of hypotheses on secret key values. These were done mainly on smart cards and micro controllers which were mainly used on the payment systems of subway, parking areas, toll booths etc.

In 2000/2001, the use of Electromagnetic Emanation as a side-channel was demonstrated by Jean-Jacques Quisquater and David Samyde [4] as well as Karine Gandolfi et al. [5]. These were the first openly published works on the EM analysis of ICs and CPUs. The EM side-channels include a higher variety of information. The analysis is performed on chip cards, and they used tiny antennas that were placed in very close proximity to the IC. This made use of the Electromagnetic radiations emitted by any embedded device and does not need a resistor in the power supply path, as is required for power analysis. It needed just a handmade loop antenna to first show that better attack could be performed with some analysis basics as required for power. It was found that space localization can be performed in electromagnetic analysis [7].

Since the EM emanations may depend on the physical characteristics of the active gates, a single EM sensor could capture multiple EM signals of different types. These signals can be extracted and analyzed individually. In general the chip contains a CPU, clock unit, memory, Flash and so on depending on the embedded system. We know that not all of them consume the same power and also not all of them are in use at a certain time. Based on the action performed, some are active and some are not at any given point

of time. So depending upon the activity, the EM radiations emitted are different and the loop antennas made can pick up these EM signals from each of the different areas of a chip. Based on the position of the antenna and chip, the EM radiation varies due to the direction of the current. This is different from power analysis wherein the power measured is the power of all the active units. Once we get different EM radiations we can find out the areas with different emissions and with the knowledge of the activity of different components of the embedded system it is easy to find out which component is where. We can also find where in the design under test are the CPU, FLASH, a RAM and an internal clock generator [14].



[14]

Figure 2.2 Floorplan of DUT used

A comparison of the EM radiations of different algorithms gives rise to more knowledge about the algorithm and with statistical data, a simple or differential EM analysis can be performed. Additionally there are EM attacks that can be applied from a certain distance. This was proven by Agrawal et al. [6] in 2002. This was the most comprehensive study undertaken in Electromagnetic Analysis, which took the unintentional emanations into consideration. Handmade near field probes were used for that and since the captured signals were very low in amplitude a wideband amplifier was

needed. The captured signal was then analyzed using an oscilloscope. The factor of needing precise measurements and better positioned probes paved way to experiments that used probes being monitored by a computer. The research in this area gained so much popularity that we now get near field probes from companies like Rhode and Schwarz, EMV Technik, ETS Lindgren.

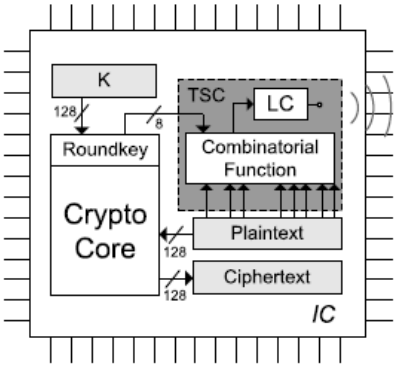
Although side-channel attacks have been specialized to custom ASICs since 1998, attacks targeting specifically FPGAs have been reported more recently. The EM side-channel attacks were performed on an FPGA that implements a cryptographic algorithm in [12], [13], [17], [18]. Thus starting from smart card, the Electromagnetic analysis diversified and instilled interest among the researchers over the years, and is being considered as a potential research area that needs to be exploited even more. This kind of near field scanning using an exclusive side-channel setup was demonstrated in [14] and [15], where even the origin of the electromagnetic emanation could be found using SEMA in the time domain. A Low-cost near field mapping system started to get used which can scan automatically and dynamically the magnetic emissions from the chip. Labs which could afford high precision equipments used one such setup in [14], where the computer controls the position of the probe over the design under test. The probes are placed very close to the design under test but even then need an amplifier, since the captured signals would be of very low amplitude. These signals after amplification are fed to the oscilloscope, which is also controlled by the computer. The computer stores the waveforms obtained and helps in the post treatment of the waveforms.

Since all these attacks were made possible it became imperative to find ways to protect them and also provide constructive uses of the Side-channel analysis. To protect

the chip from side-channel attacks, these chips started using side-channel resistant logic [21], [22], [23]. Also, there are concepts like the intentional increase in the noise so that the Signal to Noise ratio would decrease and make it difficult for the attackers, better implementation of branching, and randomness.

Recently, there have been more practices of outsourcing designs and chips instead of having them in house or home grown. This is mainly due to the cost benefits and reduced time and effort required from a company's perspective. More and more companies have started following the Intellectual property based designs and are Fab-less companies. They manufacture their chips in alien countries and thus are prone to security issues. The dangers of fabrication in alien countries are studied in [25]. The main problem of such fabrication in alien countries is that there are possibilities of insider based attacks as there is not enough control of the personnel in foreign countries. The attacker in this case can be working within the fabrication plant and plan to insert some new circuits that alter or affect the basic functionality thus breaching the security of chips. As mentioned in [48] there can be many *Phases* like Design, Fabrication, Testing ; *Abstraction levels* like RTL, Gate level, transistor level and Physical level where they can be inserted, different *activation methods* and *different places* in the design where they could be inserted. The possibilities of Hardware Trojans being inserted in chips were being thought of and even competitions to show case the usage of Trojans and possible detection of them were held [33]. Basically, the Trojans should not be found when functional testing is done. The main idea in the competition was that the Trojans remain dormant until a specific condition is reached and once it is reached it should find a way to

leak the important information needed for the attackers. Detection methods that can be used to find such Trojans have been shown in [24] and [46].



[25]

Figure 2.3 Trojan Side-channel

2.2 Constructive Use Of Side-channels: Watermarking

Due to increasing usage of different IP cores from different companies in order to build a new system or ASIC, we also need to make sure there is efficient security in place to avoid IP theft, counterfeiting and frauds. To address this issue for hardware designs, the concept of using small circuits along with the original design called *Watermarking* ([26], [29]) came into play. This is similar to the Trojan concepts explained above, but uses the same concept for constructive purposes. This idea of watermarking might be relatively new to hardware designs, but has been incorporated in the case of software codes, web page data, video and audio in order to protect the digital content. These watermarks circuits that are inserted are very small when compared to the original designs. There have been many watermarking schemes proposed especially for the software concepts where detection of such watermark needs the software code [40] or access to the memory structure ([41],[42]). However, Side-channel watermarks do not

need any such requirement, as detection of such watermarks can be possible just by monitoring the power consumption [36]. This is possible due to the very light weight structure of the watermark, which could hide itself underneath the power consumption of the other processes in the device.

The usage of EM side-channels for Hardware Trojans as well as Watermarking has not been explored, and in this thesis we have attempted to address this issue by making use of EM Trojans and modifying them as Hardware watermarks on FPGA and Software watermarks for Smartcards and microcontrollers which are explained in Chapters 5 and 6.

CHAPTER 3

EXPERIMENTAL PLATFORM

This chapter explains the experimental setup and the methods involved in performing the EM Side-channel analysis on a device under test: FPGA, Microcontroller and Smartcard.

The target device varies among the above-mentioned 3 Design under tests (DUTs), while the PC is the controlling device that controls the communication with the DUT. The PC also supplies different inputs and contains the disk space to save the data from the oscilloscope based on a trigger condition. The oscilloscope used here is Tektronix DPO 7104, where 4 channels of data can be monitored. The EM probes from ETS Lingdren are used for capturing the Electromagnetic radiations from the target device. The setup required is shown in Figure 3.1.

Choosing the best probe among the available probes can be a task that can be accomplished by testing all the probes for the best signals. After trying all the probes, it is found that the magnetic loop probes give the wanted EM radiations and are placed on the DUT after finding the best spot, by placing the loop at different locations and checking the signals on the oscilloscope.

Using the probes in different angles will produce different Electromagnetic signals with different polarity. The best possible angle is obtained by keeping the probe at different positions and checking the signal with the scope. This would be made easier when running a program continuously and checking the scope by changing its settings as and when required. In general, the smallest loop probe available should be used so that

the advantage of locality can be fully achieved and the probe in a slightly tilted position as shown below, gave us better results.

Figure 3.1 **Measurement Setup for EM Analysis**

3.1 Hardware Used

3.1.1 Target Devices

We used FPGA, Microcontrollers and Smart Cards as target devices. We have used the smartcards and microcontrollers as they are increasingly used in day-to-day life and can be a perfect platform for security related research. The FPGA has been used since it is relatively easy to obtain when hardware chip is concerned. Custom made ASICs could be made but require more effort and time. Although, FPGA would involve considerable amount of hardware design and implementation skills and is a preferred target for current hardware security research.

In FPGA, we have mainly used Spartan 3E Starter Board as our target device but we have also used a Basys2 board from Digilent in the case of hardware Trojans, where

we used the Basys2 board as the Crypto Core and Spartan 3E Board as the Control Core. Implementation of logic designs like Toggler, PRESENT, and TEA for FPGAs is done in Verilog. We wanted to test the watermarks in the case of smartcards. So we had to get a smartcard which could be written into and got a microcontroller based smartcard. We made use of an ATMega 163 Fun card. We also need a Smartcard programmer for which we used Dynamite + Plus programmer and SCM 3310 reader. We have also used an ATMega 8 microcontroller.

3.1.2 **EM Probes**

The EM probes used in this thesis are ETS Lindgren 7405 Near-field EM probes. We have mainly made use of the magnetic probes and in particular 903 Magnetic (H) Field Probe which has a loop of 1 cm. The E/H or H/E Rejection is about 11 dB and it can support up to 2.3 Ghz of resonant frequency. Since EM signals are prone to noise and can be easily distorted in a very noisy environment like a lab, it is necessary to use a pre-amplifier before connecting the EM probe to the scope. We use a pre-amplifier which has a bandwidth of 100 Khz to 3 Ghz.

3.1.3 **Oscilloscope**

The Oscilloscope used here is Tektronix DPO 7104, which has a bandwidth of up to 1 Ghz and has 4 channels with a maximum sample rate of 40 Gs/s.

Apart from these main devices needed, we would also need a GPIB or LAN connection in order to capture the traces from the scope. We have used Agilent 82357B

USB/GPIB Interface for FPGA, and LAN for Smartcard and micro controller, to provide Communication between the Oscilloscope and the PC.

3.2 Target Platform Basics

The first step for all the analysis is to get the platform up and running. Different DUTs require us to know different functions, protocols and methods. This section explains how to make the different platforms ready for data acquisition and analysis. This is to mainly explain what equipments are required, and what steps are to be followed to get the setups created.

3.2.1. Smartcards

We work with smartcards to insert watermarks and detect those using EM side-channels. The first thing is to choose the smartcard and smartcard reader that communicates with the smartcard. We have used the *ATMega 163* Card, which has an 8-bit microcontroller with 512 Byte internal EEPROM and 16K Byte programmable flash memory and 256K bit external EEPROM. This provides enough space for programs that include crypto blocks as well. We have written the watermark program in assembly level language using *AVR Studio*. The crypto OS and the functions are obtained from an open source code available online. We will need a programmer that can program the respective smartcards. We have used *Dynamite Plus smartcard programmer* with its compatible software: *Cas Interface Studio 9.0.0*. This software can program a number of other smartcards as well. The procedure to write a program into the card is fairly simple and

can be obtained from the Cas interface user manual [43]. This explains what we need to write the assembly codes and program it onto the smartcard.

Once we have the smartcard ready, we will need a smartcard reader that is controlled by the PC to communicate with the smartcard by sending messages and waiting for the responses. We should understand how the open source code uses the transmission protocols and the different functions so that we can write the wrapper for the smartcard programmer which communicates with the smartcard and the PC. The aim of our wrapper program is to send messages to the smartcard, wait until it performs the required code with watermark and save the responses. This is done a number of times while the EM probe can capture the EM traces from the smartcard.

The contact type smartcard falls under the ISO/IEC 7816-3 standard. There are two main transmission protocols T0 and T1 we will need to know. The difference between the 2 protocols arises from the way data is transferred, where T0 is byte oriented and T1 is block oriented. The error detection used for the transfer also varies with the T1 protocol having an EDC at the end of each block apart from the parity bits which is common for both the protocols. The open source protocol uses T1 protocol in our case.

The protocol used for transmission is important as it is based on the protocol that the APDUs (Application protocol data unit) are designed upon. The APDU is required for the entire data exchange between the PC and the smartcard. But APDUs which are under the ISO/IEC 7816-4 standard are independent of the transmission protocols. As we said earlier, we have to send messages and receive responses from the smartcard. Both of these follow a certain structure, and are listed below. This should be known in order to

send and receive messages properly. We will make use of this structure while we write the wrapper for PC and smartcard communication.

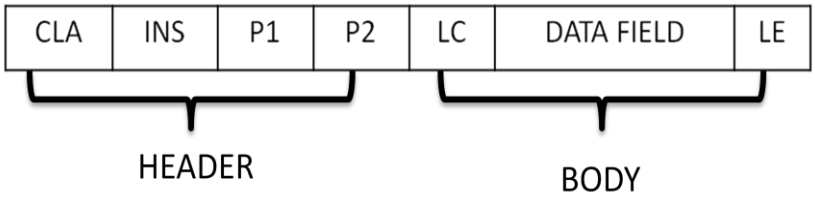


Figure 3.2 **Structure of a Command APDU**

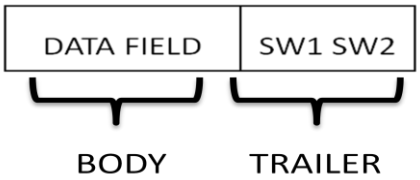


Figure 3.3 **Structure of a Response APDU**

The following table denotes what each field means in terms of the card communication.

Table 3.1. APDU Description

Field name	**Description**
CLA	Instruction class. Used to Identify applications and their specific command sets
INS	Instruction byte which encodes the actual command. In our case, it can be used to denote Set Key, Encrypt and Decrypt
P1-P2	Instruction parameters for the command in order to provide more information about the command. Eg. Offset into file at which to write the data
LC	Indicates the number of bytes of command data to follow
DATA	The actual data being sent
LE	Indicates the number of bytes of data sent back from the card

The card will always send the trailer back, and the body is optional. The body is the response data, dependent on the operation being performed. The two bytes SW1 and SW2 are called the return codes, and denotes the response from the card. The SW1 and SW2 can assume different values based on how the communication proceeds. The value varies for successful communication and a failure. The causes for a failure in the communication can also be found out from this return code. Knowing the causes for failures will help in debugging the errors. The main categories of return codes are listed below in the Table 3.2. For more exclusive error codes, refer to [44]. The following table denotes the different return codes and their descriptions.

Table 3.2. Return codes and description

Return Code	**Process**	**Process Completion**	**Description**
90 00 61 XX	Normal Processing	Process Completed	Success
62 XX	Warning Processing	Process Completed.	Warning: State of the non volatile memory not changed
63 XX	Warning Processing	Process Completed.	Warning: State of the non volatile memory changed
64 XX	Execution Error	Process Aborted	Error: State of non volatile memory not changed
65 XX	Execution Error	Process Aborted	Error: State of the non volatile memory changed
67 XX To 6F XX	Checking Errors	Process Aborted	Check errors

We have now discussed smartcards and the transmission basics. The last part of making the smartcard setup work is to make use of the programmed card and the protocol basics, APDU formats, and create a wrapper or a program that automates the sending and receiving of messages between the reader and the smartcard. We have used an SCM 3310

smartcard reader. This is a PC/SC compatible smartcard reader that can work with T0/T1 protocols. The PC/SC framework is the necessary OS required for the smartcard environment to be run on a PC based system. Microsoft has created a set of functions that should only be used in order to work with the PC/SC based smartcard reader and cards. There are a number of functions listed in [45] but we can make the smartcard environment work with just the basic required functions. These are listed below.

1. SCardEstablishContext
2. SCardListReaders
3. SCardConnect
4. SCardTransmit
5. SCardStatus
6. SCardDisconnect
7. SCardReleaseContext

SCardEstablishContext is the first function to be called. This function is necessary to establish the Resource manager context within which all the other operations can be performed. This is the platform that needs to be established or connected so that the smartcard environment is in use.

SCardListReaders is the function that lists the various smartcard readers connected to the resource manager context. We use only one smartcard reader but this function can list all the readers that are connected. In our case this function would return SCM 3310 Microsystems.

SCardConnect is the function which when called, can establish the connection between the calling application and the smartcard which is inside the smartcard reader using the resource manager context.

SCardTransmit is the function that is used to send some commands to the smartcard and expects to receive the response from the card. This function makes use of the data to be sent as a parameter variable but it should be made sure that the command data follows the APDU format explained above. This function could be used in a loop to make the setup automated.

SCardStatus is the function that is needed to check for the status of the connected card and reader, to find out whether the card is connected properly. You can call it any time after a successful call to SCardConnect and before a successful call to SCardDisconnect. I have used this function all the time so as to find the status of the card and reader at all times. When there is no problem the function need not return anything. But in case of a disconnection we could have an error that pops up.

SCardDisconnect is used when all the necessary transactions are completed. This is to disconnect the existing card from the resource manager context. This can be used if we need to use more than one smartcard. It should also be noted that in that case we need to use SCardConnect for the new card after disconnecting the existing card.

SCardReleaseContext is the final function that should be used to release the resource manager context. It frees up all the resources that were used and also frees up the memory used. Once this is called, we need to start from SCardEstablishContext in order to use the smartcard system again.

I used the above mentioned functions and automated the transmission of messages in order to get the EM side-channels for a number of measurements.

3.2.2. FPGA

The other platform we used in this work is based on FPGA. We used it for Simple Electromagnetic Analysis, Differential Electromagnetic analysis, EM analysis for hardware Trojans and Watermarking. The hardware needed for this is the FPGA being used. We have used a Spartan 3E FGA and also a Basys2 FPGA. We have also de-soldered the coupling capacitors in order to help for power analysis in the case of Spartan 3E FPGA so as to isolate the noise sources from the board.

We write the codes for the FPGA in Verilog and use Xilinx ISE for all the work related to the design. We generate the HDL, synthesize it, run the timing, and place and route the design using Xilinx ISE. We then port the bit file using Impact function of Xilinx ISE for Spartan 3E, and Digilent Adept in the case of Basys 2 FPGA. We also use Xilinx EDK in order to create a microprocessor inside the FPGA. This is used in the case of Trojans and Watermarking circuits for the FPGA that controls the crypto core. We need this design to send different Inputs and a Trigger signal. The reason why we keep this as a micro processor and in a separate FPGA is to avoid the noise that can interfere with the crypto core which we will need to attack and detect the secret value. The connections between the 2 different FPGAs are connected using normal wires that ensure connection between the IO pins.

3.2.3. Microcontrollers

We have made use of an ATMEGA8 microcontroller, which helps in creating a very light-weight watermark using only 4 to 5 assembly level instructions which can easily be hidden in the rest of the operation the microcontroller performs.

We created a 2 PCB setup using the ATMEGA8 microcontroller in one of the Printed Circuit Boards (PCB) and used the other PCB to supply the different challenges using a serial connection. Both were powered with different power supplies. The 2 PCB setup also isolates the core which here is the microcontroller from outside sources of noise such as the fluctuations of power supply from a PC and capacitances in the power supply line. We can use AVR Studio to create assembly codes for the micro controller. Apart from this, we will need codes to automate the capture of traces, and once we get the traces, codes for analyzing them.

3.3 Steps In Acquisition Of Data

For performing successful side-channel analysis we need to obtain a lot of measurements from the target device. We need this by doing a certain procedure a number of times by automation. The steps needed to be followed are:

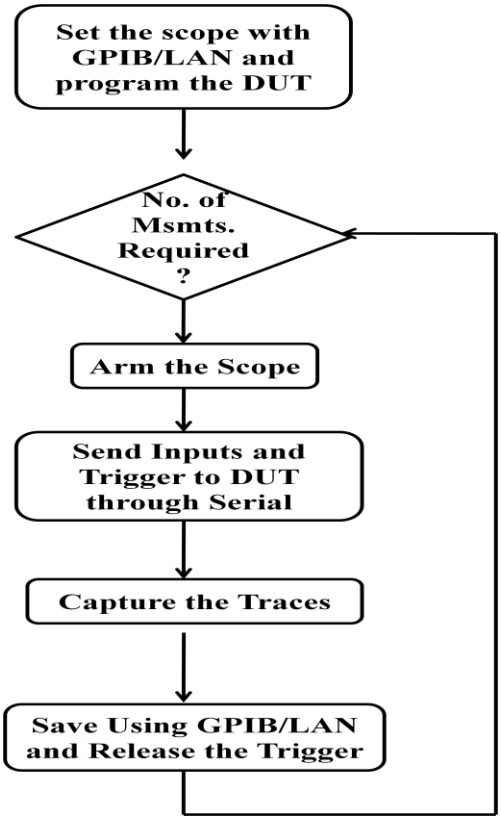


Figure 3.4 **Data Acquisition Steps**

As seen in the above flowchart, we first need to establish that the desired target device has the desired logic and is waiting for a trigger and inputs. The PC is used to control the Serial Connection needed to communicate with the DUT so that each new set of inputs and a trigger signal are given. In order to capture the traces for different set of inputs, the oscilloscope should be armed in such a way that when there is a trigger, the oscilloscope waits and captures the traces obtained by the probes. This captured trace should be then stored in the PC. This is done by using GPIB or LAN. Once the trace is captured, the trigger option in the oscilloscope should go back to the normal state so that the next trigger can arrive and the new set of traces can be obtained in the same manner as described. This procedure is done until the required number of measurements is done.

3.4 Pre-Processing Of The Traces

The traces are captured based on a trigger signal from the target device. Finding a trustable hardware trigger signal is a difficult task and, even if this is accomplished, there can be some errors as these hardware generated triggers are bound to have some mismatch and would not arrive instantly as required, since the clock signal of the target device by itself is not without errors. For example, The Spartan 3E board uses a 50 MHz clock generated by a crystal oscillator. But this clock signal when measured varies for different Spartan 3E boards. It was 48 MHz for one board and 52 MHz for another. Therefore, some margin of error has to be allocated for this and this necessitates pre-processing of the captured data before performing post-processing analysis to extract the key.

3.4.1 Alignment

Sometimes the algorithm does not start at the same time. This can affect the correlation of the signals as there are 1000s of traces being captured and each can be different and might not have the expected peaks at the desired location. The following diagrams clearly show how different traces have different peaks and how it affects the overall traces.

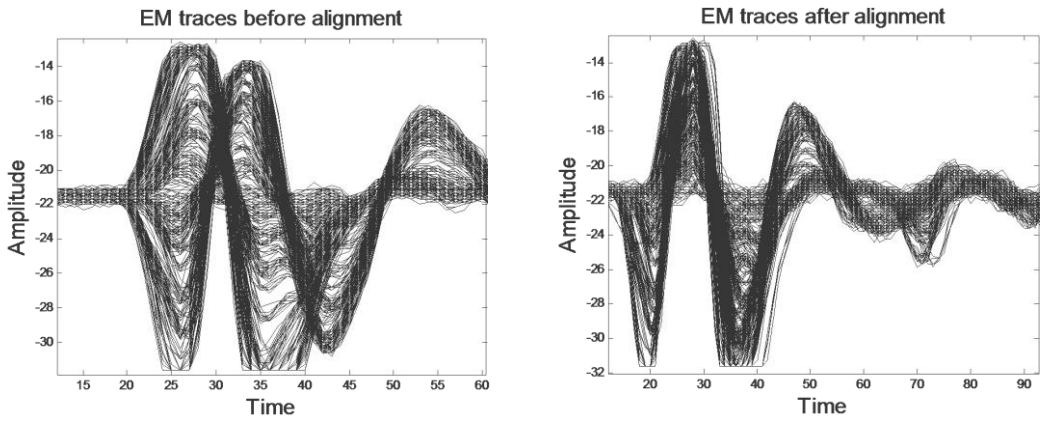


Figure 3.5 **a) Traces Before Alignment b) Traces After Alignment**

Though the amplitude difference between the two trigger traces is not that big, the point to be considered here is, there will be differences in the timing period, which needs careful alignment. This could be because of clock jitters, or if two different cores with inevitable minute differences in the clock rates or altogether different clock rates are used. This is shown in the graphs before and after alignment. In Figure 3.5-a, we can see the accumulation of errors in the positions of a lot of traces. Figure 3.5-b shows the same set of traces after alignment. This step of alignment is the first step to be followed when considering any attack.

3.4.2 Filtering

Filtering the traces for required frequencies and leaving out the unimportant frequency component of the traces can improve signal to noise ratio in terms of correlation. The unimportant frequency components are considered to be noise since it does not provide any use for the EM analysis. We will know the frequency at which the FPGA or smart card runs. As mentioned in [6], there can be some information from the unnoticed frequency components. If we know the range of needed frequencies, we can

use a Bandpass filter on the raw traces before using them for EM analysis. Also, if we need only lower frequencies/ higher frequencies we could use a Low pass filter/ High pass filter. Filters can be easily generated using the fdatool of Matlab. It involves a GUI wherein we can enter the frequency of interest and generate codes for such filters.

The following diagrams show one such trace obtained from the Spartan 3E FPGA.

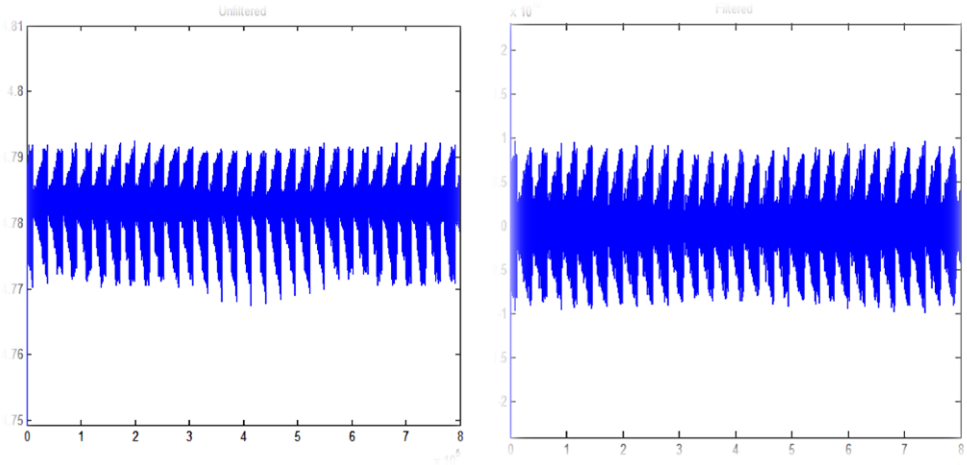


Figure 3.6 **a) EM signal with low frequency component domination. b) EM signal After Filtering**

It can be seen that Figure 3.6-a shows a continuous stream of toggle operations at high frequency but it can also be noted that it follows a low frequency modulation signal. When we do not need this low frequency modulation signal, we could remove it using a high pass filter and using those low frequencies in the stopband of the filter. Figure 3.6-b shows low frequency removed and how it can flatten out the toggle waveform obtained. Filtering will be useful when we collect a number of traces with large sample points.

3.4.3 Frequency Domain Analysis

All the above work was done in time domain. In some cases ([6], [8]), it is proved that the frequency domain is used to suppress the unwanted noise and also to reveal more information from the normally neglected frequencies. As it is found that sometimes the third harmonics can contain important information about the key, changing the domain of the traces helps in identifying which frequency components are important. It is found from the previously mentioned works that not only in the frequencies that are multiples of the clock, but also in the other low frequencies, potential side-channel information could be leasked. Such EM analysis is called Electromagnetic Frequency Analysis and has been proved successful although in this thesis all the work was carried out in the time domain.

CHAPTER 4

SIDE-CHANNEL ANALYSIS ON FPGA

This chapter explains the implementation of Side-channel attacks on a Spartan 3E FPGA board. Electromagnetic side-channel analysis is done on the implementations of "Toggler" and "AES SBOX" designs.

4.1 Programs Used

4.1.1 Program1: Toggler

The Toggler design is a small logic which has a 20 bit register that has values with increasing hamming weights every alternate clock cycle separated by a return to zero so that Electromagnetic analysis could be done on the hamming distances. Once the value reaches 20 logic 1s, it starts back again at 0 moving towards 20 1s. Simple Electromagnetic Analysis (SEMA) is employed on this design. Table 4.1 depicts the operation of the Toggler.

Table 4.1. Operation of Toggler

Clock Cycle	Value of the 20 bit register
1	0
2	1
3	0
4	11
5	0
6	111
….	…..
40	11111111111111111111(20 1s)

41	0
42	1

4.1.2 Program2: AES Sbox

This is an implementation of an AES Sbox with different inputs having three main parts which are listed below

1. LFSR generated different 8 bit inputs
2. XOR function of 8 bit input and 8 bit key
3. AES Substitution Box

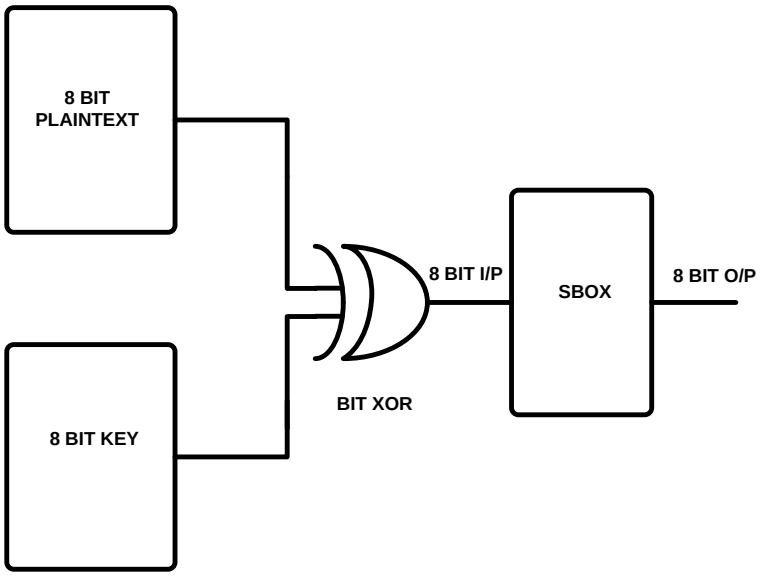


Figure 4.1 **Components used for DEMA on AES SBOX**

This implementation is used to test Differential Electromagnetic Analysis (DEMA) on the Spartan 3E Board. The initial seed for LFSR is noted so as to perform the Analysis after capturing different EM and Power traces. The XOR function of each 8-bit

input and 8-bit key forms the main crypto design in this implementation along with the Sbox table. The analysis is done to extract the correct key guess among the possible 255 key guesses.

4.2 Simple Electromagnetic Analysis

This type of side-channel analysis is generally used to extract the key of a cryptographic design by just capturing a trace and locating the operations, and based on clocks, guess the key mainly based on visual inspection. We have made use of this technique for a simple toggle program so that we can easily visualize the operations involved, even though there is nothing crypto in this implementation. From the diagram given below, it can be observed that the operation followed by the design is easily detectable by visual inspection.

From the logic listed above, we can see that the hamming distance keeps increasing before falling to 0 so that the difference of hamming weights and hamming distances between 2 consecutive clock cycles is always the same. What we expect is to find a waveform that looks like saw tooth waveform. The EM and power traces are captured when the FPGA is functioning. The following diagrams depict the Clock and EM signals obtained from the Spartan 3E FPGA.

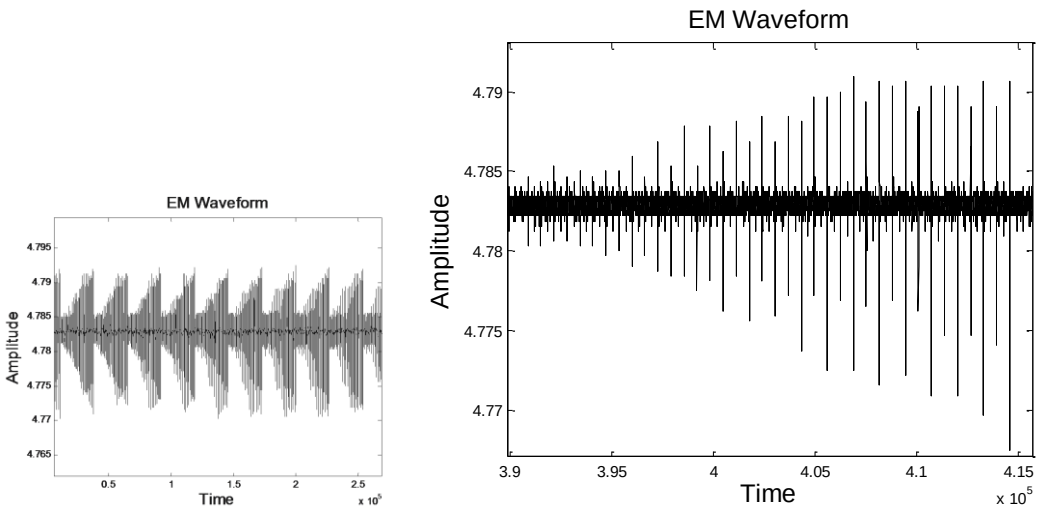


Figure 4.2 **Toggler EM Waveform**

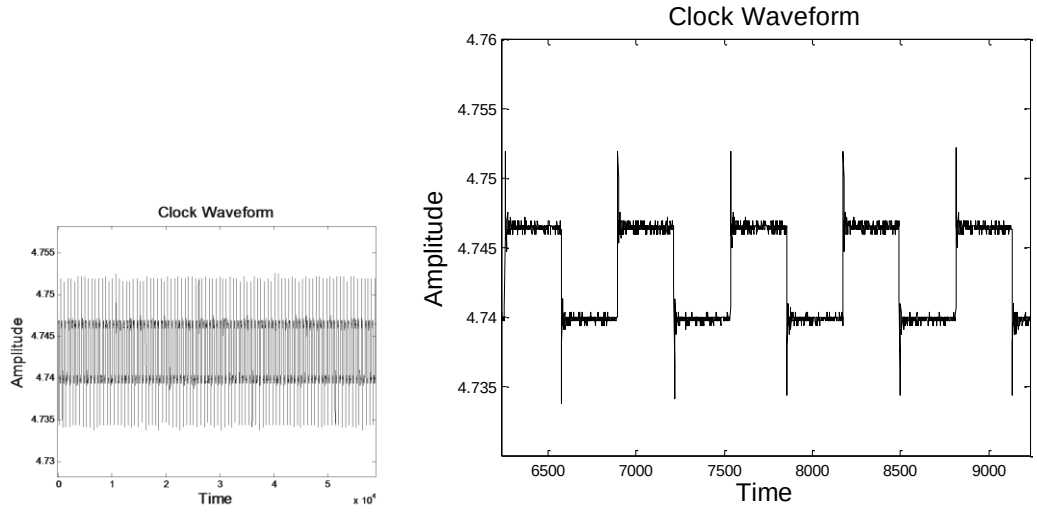


Figure 4.3 **Toggler Clock**

Once we have the waveforms of clock and EM radiations we inspect the number of clock cycles needed for one single toggle operation to complete. We note down the clock cycles at which we have the peaks from the EM operation. These peaks should coincide with the time at which each step in one toggle operation occurs. The Hamming distance model is also predicted, and for toggle the prediction is easy, as the hamming distance should keep increasing for this operation. In this case HD (v_0, v_1) = HW (v_0, v_1).

The comparison between the peaks and the corresponding hamming distances are found out using Matlab. We make use of this comparison for all the possible peaks and predicted hamming distance models. This way, our prediction was correct almost 95% of the time as the hardware implementation of a very small design is easier and does not have many errors.

The same mechanism can be used for power analysis which will require a resistor in the path of the power supply VDD of the FPGA chip. Using the scope's cable to monitor the power or current across the resistor gives the power waveform. This does not involve key extraction but to get ourselves familiar with prediction and comparison of obtained data, this experiment helps. This method, when implemented on a key dependent design should be able to reveal the key using Simple electromagnetic analysis as in the case of Square, Multiplication operations of RSA shown in the Introduction chapter.

4.3 Differential Electromagnetic Analysis

This side-channel analysis is complex and is used if the visual inspection and direct relationship between the waveforms and the key cannot be known. This method involves a sequence of steps in statistical analysis that needs to be performed repeatedly in order to retrieve the key. Almost all attacks need this type of analysis nowadays as there is a considerable amount of security in place in embedded systems. The following figure explains the procedure required in Differential analysis.

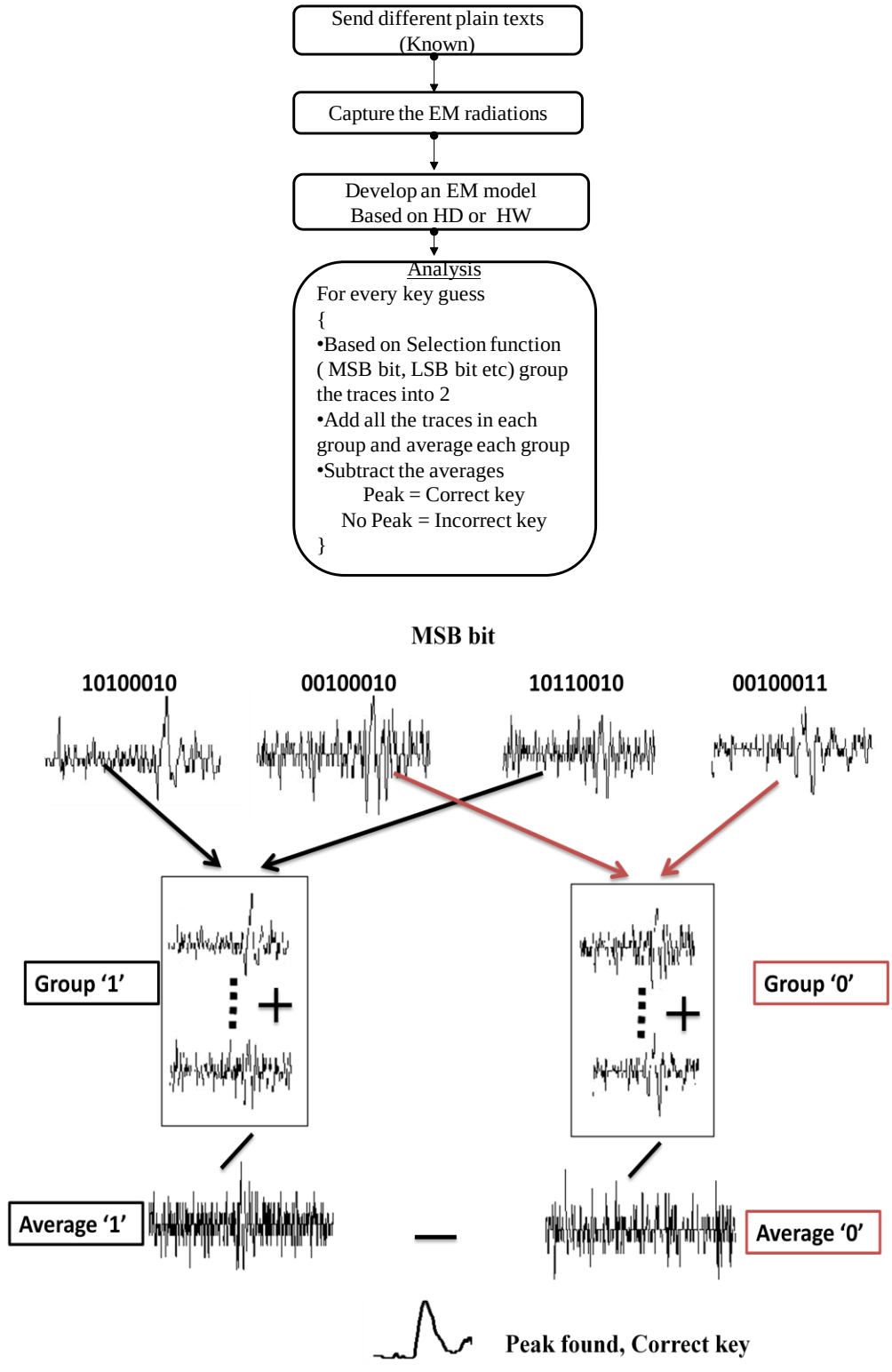


Figure 4.4 **Differential Analysis Procedure**

We use this procedure for our AES S box implementation and the steps used are explained below:

Step1:

Select the algorithm with a secret key and enable it to accept different inputs. The measurement setup is made in such a way that it is possible to know the different inputs during the analysis stage. For this purpose an LFSR is used to provide the inputs to the AES Sbox and we also know the seed value so that we can simulate or predict what the values are for each clock cycle.

Step 2:

Selection of the target signals. In general many algorithms will use a number of Sboxes and the EM side-channel attack should be made on each and every Sbox to find out the entire key. In our case, we use a single Sbox as the target. We target a bit of the Sbox output which performs the XOR of the inputs obtained from the LFSR and the 8-bit Key.

Step 3:

Measurement of different inputs is obtained using the measurement setup. In our case around 4000 traces are obtained.

Step 4:

Prediction of the Sbox output which we have as the target is made. Here the combinational function is an XOR operation between the inputs and 8-bit key. Since we know the initial seed we can find the predicted outputs based on the combinational function we use.

Step 5:

The prediction of every key guess (255 in this case as 2^8) is carried out in a loop with every different input and, with the knowledge of the different inputs, the EM radiation model is generated.

Step 6:

Once the EM waveforms are measured with different key guesses, they are separated into 2 groups: ‘group 0’ or ‘group 1’ based on the guesses of the key bit. Then the waveforms in group 0 are all added and averaged. This is done for group 1 too. The the difference of the average of the 2 groups is then calculated. This leads to a plot with just one key guess being correct which could be distinguished based on the correlation value plotted.

These steps are followed in a loop with the n being 255. This varies with the number of key bits considered. Generally this is done in terms of 8 bits or 1 byte of the key and continued for all the key bytes.

Figure 4.4 shows the correct key guess having peaks better than the others. The correct key guess is noted and if it is for a single S-box, it is then carried out for the other S-boxes as well. This method in general can be applied phase by phase to reveal the entire key or can be used in order to reduce the efforts needed by a Brute force attack.

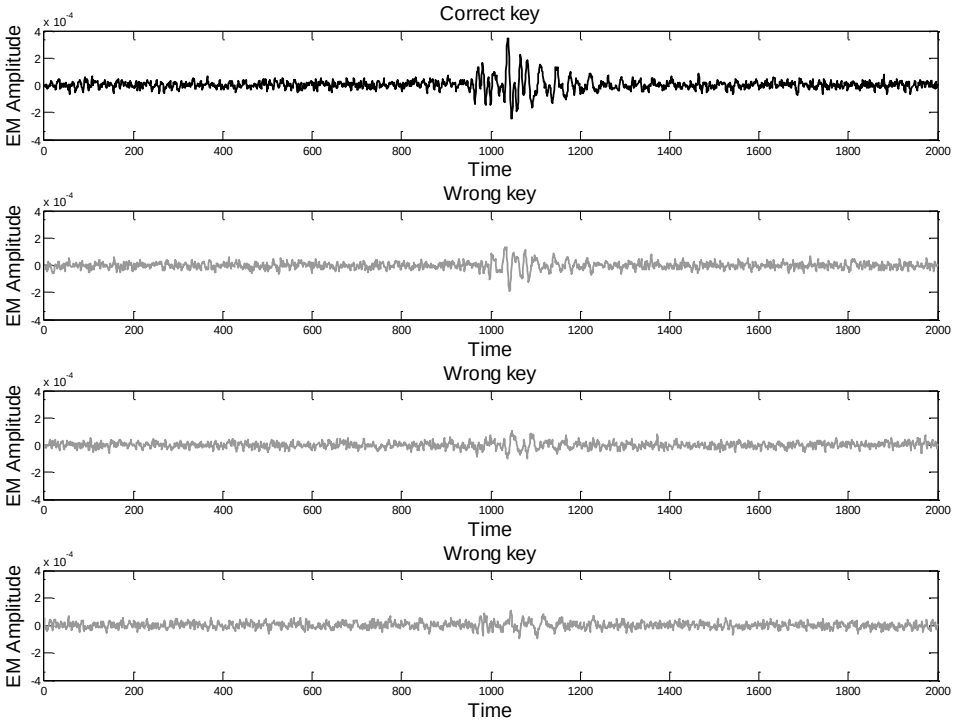


Figure 4.5 **DEMA curves with correct and wrong key guesses**

4.3.1 Correlation Based EM Analysis

An efficient method that was developed as an improvement of DEMA was the Correlation based EM analysis. Here the difference is that there are no grouping, averaging and difference steps. Instead, this analysis is based on the correlation of the hamming distance models and the obtained traces. The model could be Bit model, or hamming weight model as well. This method also focuses on a few bits at a time instead of just 1 bit as in the case of DEMA. The prediction of the algorithm, key guesses is the same as DEMA.

Here correlation between 2 points is considered. The 2 points are actual EM radiation and the hypothetical EM radiation obtained after following the steps 1 to 5. It is given by the formula

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X).Var(Y)}}$$

where, $Cov(X,Y) = E(\big(X - E(X)\big).\big(Y - E(Y)\big))$

$$Var(X) = E(X^2) - (E(X))^2, Var(Y) = E(Y^2) - (E(Y))^2$$

E(X) is the mean.

After finding the correlation between different key guesses, a CC plot is plotted, which has the highest correlation for the correct key guess as shown below. In the figure, the trace in black is the correct key, while all the grey traces are wrong key guesses. We can see that the correct key clearly isolates itself from the incorrect key guesses.

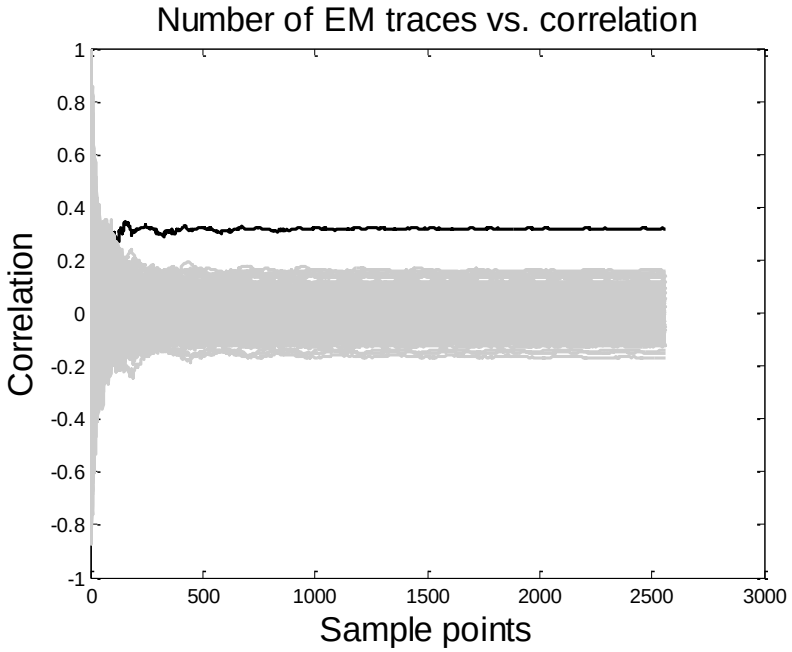


Figure 4.6 **CEMA Plot**

CHAPTER 5

ELECTROMAGNETIC ANALYSIS USING HARDWARE TROJANS

This chapter explains the work of hardware Trojans on FPGA. It also provides background information on the Symmetric key algorithms that are used in this thesis.

5.1 Algorithms Involved

5.1.1 TEA-Tiny Encryption Algorithm

The TEA- Tiny Encryption Algorithm [34] is a block cipher which is noted for its simplicity and implementation which uses only very few lines of code. It was designed by David Wheeler and Roger Needham in 1994. Since this algorithm is very simple in implementation it found its use in a lot of day-to-day applications such as Xbox consoles.

TEA operates on 64-bit blocks and uses a 128-bit key. It has a Feistel structure with a suggested 64 rounds, typically implemented in pairs termed cycles. It has an extremely simple key schedule, mixing all of the key material in exactly the same way for each cycle.

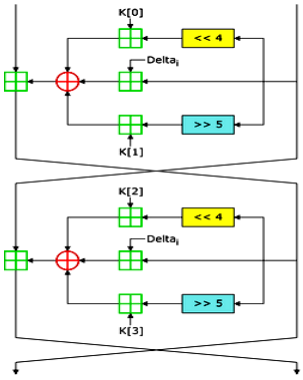


Figure 5.1 Two rounds of TEA

5.1.2 PRESENT

PRESENT is a 64-bit block cipher with an 80-bit key, designed for low power, low gate count applications. In order to evaluate their design, the creators of PRESENT normalized several area optimized designs to the NAND2 gate of the technology used, a technique known as gate equivalence (GE). The authors claim an approximate GE of 1.5k in their implementation of PRESENT.

The basic PRESENT algorithm is given in Figure 17 from [35]. There are 32 rounds with each round consisting of a 64 bit round key XOR, S-box, and a permutation layer. After each round, the next round key is generated by the key scheduler and is represented by the update box in Figure 5.1. The S-box is the direct 4-4 bit substitution. One of the powerful aspects of PRESENT is the simplistic manner by which the PRESENT S-box can be implemented in hardware using only 10's of gates, which compares favorably to AES 8-bit S-box that requires approximately 200 gates. The permutation layer reallocates 62 of the 64 bits to different bit positions. Naturally, this layer does not require any gates, only specific wire routing, and does not contribute to the GE of PRESENT.

5.1.2.1 PRESENT Key Scheduler

Before each round of S-box and permutation, an XOR is done with the 64 most significant bits of the key, which is then updated with key scheduling algorithm. This is done a total of 32 times. The key scheduling algorithm states that the key is rotated by 61 positions, the 4 most significant bits are put through the PRESENT S-box, and then an XOR of bits 19-15 and the round counter is done.

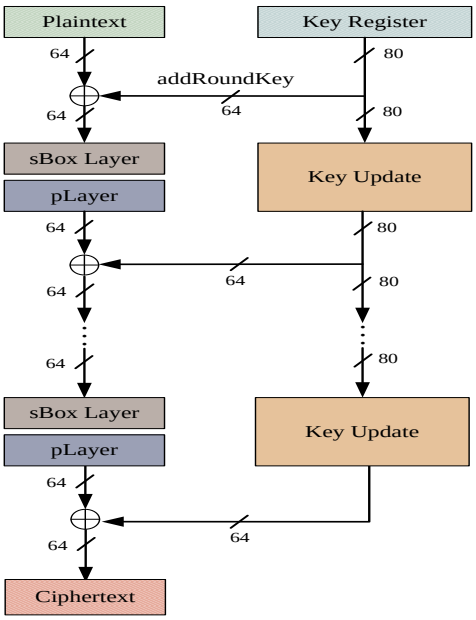


Figure 5.2 PRESENT Algorithm Block Diagram

5.2 Hardware Trojans

A Hardware Trojan is a malicious modification of the circuitry of an integrated circuit. It is generally characterized by how it is represented physically and what it does when needed. In general, malicious Trojans try to bypass or disable the security fence of a system. It can leak confidential information by side-channels like power, EM. The hardware Trojans can be designed in such a way to make the chip malfunction at important times, or leak some important information when it is related to crypto core or military devices or could even make the chip unusable or damage it permanently. Hardware Trojans can be any logic function or circuit primitives, but the most dangerous Trojans are the ones that can evade most formal Trojan detection techniques.

5.2.1 Short Background

Today's business is global and for this reason outsourcing tasks is a common method to increase companies' revenues and cut down on unwanted employee hours. That is why embedded hardware devices are produced abroad. But along with profits to the companies come serious threats, especially for government agencies. Mainly the companies and departments that involve classified information like the military, finance, and health-related companies need to be vigilant about hardware security. The hardware integrity, i.e. a chip has no modifications in comparison with the original chip design, is not ensured. Anyone that has access to the manufacturing process of a chip can do malicious alterations to the design. The Trojans could be very light weight as in the case of [25], making use of only a few transistors when compared to the original design. The fabrication of integrated circuits that are manufactured in untrustworthy factories is common. In case of an attack scenario, an adversary tries to hide the additional components; hence advanced detection techniques are necessary and are being looked into.

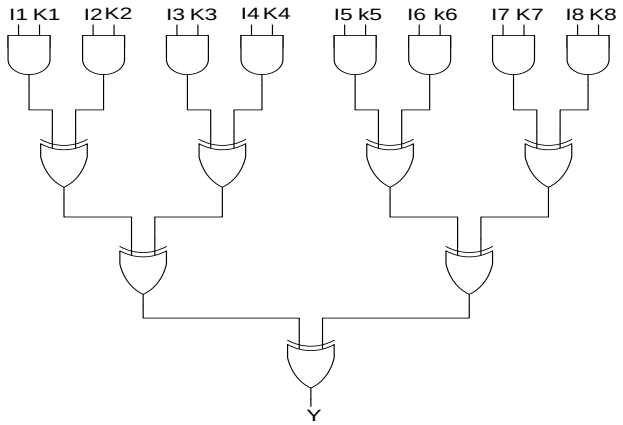
5.2.2 EM with Hardware Trojans

This work is an extension of the work [47] that I along with other lab mates did for Embedded Systems Challenge 2010 [33], which is a Hardware Security related Competition where we were given a design implementing an encryption algorithm hardened against detection schemes and asked to embed Trojans in it so as to leak key or plaintext bits.

The design used is an encryption algorithm called Tiny Encryption Algorithm which accepts a 64-bit plaintext inputs and computes a 64-bit cipher text with an 80-bit key. The Trojan was embedded in such a way that it will leak the key through EM radiations off the chip.

5.2.2.1 Implementation

Our Side-channel Trojan consists of two components: a combination function and a leakage generating circuit. We made use of a combination function that takes eight input bits and eight bits of the key to generate a one-bit output. In our implementation we used a simple combination function that ANDs eight key-bits with eight input-bits and then computes the XOR sum of these values:



[47]

Figure 5.3 Combination function for Trojans

This output bit Y is then leaked out ("transmitted") using the leakage generating circuit. The function of the leakage generating circuit is to generate high activity and thereby high EM radiation at a particular point of time, i.e. when the output of the combination function is 1 and low activity when the output is 0. We realized this using a

few 10-bit shift registers initialized with alternating 1s and 0s that are clocked in the case where output is 1 and are not clocked if the output is 0.

This set up can leak out eight bits of the key. To be able to leak out the entire 128 bits of the key we subsequently add all of the 16 key bytes to this combination function.

5.2.2.2 Activation

The Trojan is activated when it receives the inputs and a trigger signal. We used eight input pins to read in the input bits and an input pin as a trigger to activate the combination function. It is also possible to use the switches and buttons of the Basys2 board instead. However, we chose input pins since it made automated measurements easier.

5.2.2.3 Trojan Data

To reveal the key we performed a differential electromagnetic analysis just as it is done in traditional side-channel attacks. In the first step, we took EM-traces of a lot of measurements (10,000) for different inputs. We aligned each of these measurements and stored the EM-trace and the used input value for each measurement.

In the next step, we performed an EM analysis to reveal the correct key. At first we created a hypothesis for all the measurements for all the possible eight-bit keys. To do this, we used the input value for each measurement and computed the output of the combination function for all 256 possible eight-bit keys and stored the result in an array. This gave us 256 different hypotheses, one for each possible eight-bit key. In the last step

we correlated each of these hypotheses with our measured traces. The correct key should generate the highest correlation.

5.2.2.4 Experimental results

We performed the side-channel analysis as described above using the above mentioned Trojan setup but we found that the EM measurements were very noisy and in order to increase the signal to noise ratio, we connected the output to an IO pin and used the waveforms at that IO pin. Figure 2 shows the result of this Trojan work. For this analysis we set the key to "F8F8F8F896969696C3C3C3C3AAAAAAAA" (in hexadecimal). As can be seen in the picture, we were able to reveal each key bit. Although, it should be noted that we had to feed the output of the combination function to an IO pin in order to increase our Signal to Noise ratio. Though this cannot be an actual attack scenario in real world, this is a proof of concept that the Trojan analysis could work if there is better Signal to noise ratio or in other words, decapsulation of the IC or dedicated rooms for attacks without interference of noise will definitely yield better results even without the IO pin connection. Figure 5.4 shows the correlation of the 10,000 EM-traces with the possible keys. The left figure shows the correlation for each key guess. The four keys "96", "AA","C3" and"F8" can easily be identified by the four correlation peaks. The right picture shows the correlation for different timeslots.

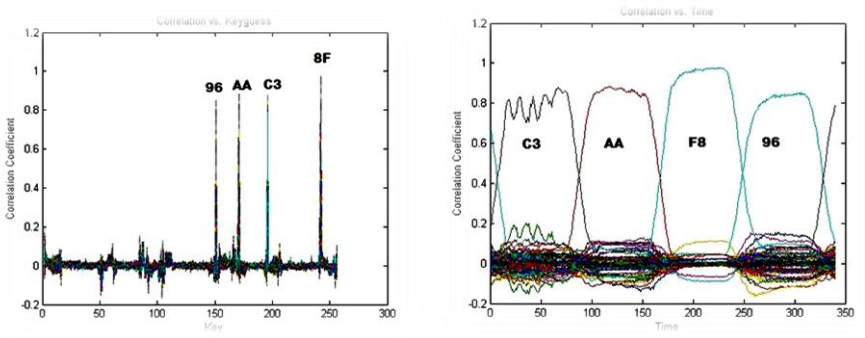


Figure 5.4 Leakage of the key with the help of an IO pin

The following figure shows the setup where we used 2 FPGAs with Spartan 3E as the "Control FPGA" giving in Inputs and trigger and Basys 2 FPGA As "Cryptographic FPGA".

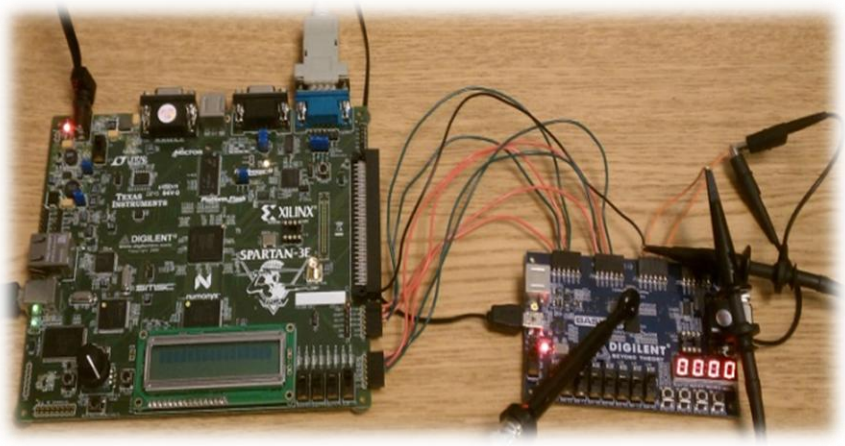


Figure 5.5 Trojan Setup with Spartan 3E as Control FPGA and Basys2 as Cryptographic FPGA

The same kind of analysis was done for one more algorithm to find out the correlation co efficient with the key set to all 'FF'. However in this case, the number of registers that were used for leaking out was increased to 15. We can see that the ease with which it could be attacked increased from the fact that the correlation coefficient increases to be more than 0.9. For the second algorithm which in this case was

PRESENT, with better implementation and some resistant techniques like having a lot of dummy logic to decrease the SNR of EM signals, it was also found to be sufficient for a successful attack with a correlation coefficient of 0.7.

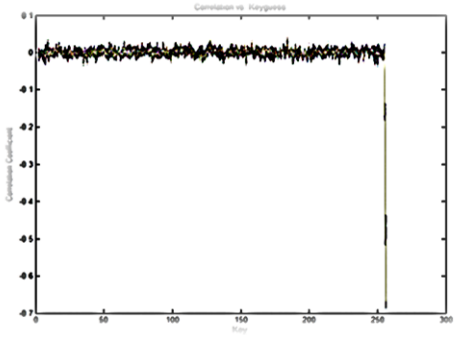


Figure 5.6 EM Analysis on PRESENT with more registers to leak

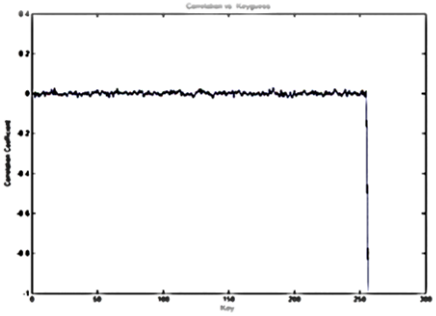


Figure 5.7 EM Analysis on TEA with more registers to leak

CHAPTER 6

WATERMARKING

In this work, we focus on IP theft where in the IP from a legitimate owner is being claimed falsely by another party. Since EM analysis needs only a probe and can detect the watermark passively without the need for monitoring the power supply line as in the case of Power Analysis, it can be used both Pre and Post production stage. Our method can help in performing EM analysis with Automation to verify the original design in the case of smartcards, microcontrollers, etc before the mass production stage, as well as help the vendor provide proof of originality to a customer at the counter after the production stage. This work also compares the EM and Power Side-channels watermark detection at different levels of setup – clean, custom made setups without noise interference to setups under external noise. It should be noted that this work will use designs that were used for the Trojan work [47] and modify and use them for constructive purposes. But there is also one scenario where one can sell counterfeit products under somebody else's brand name. The main threat is to the company as its brand name will definitely get a hit as these defective or lesser quality ICs will definitely wear out before the expected life term of a better brand IC. In this thesis we do not focus on this scenario as only the owner of the watermarks we consider can prove it is present. Thus we focus only on IP theft in this thesis and not IP counterfeiting.

Since EM Watermarking has never been attempted before, this provides motivation for EM watermarking and we propose to use EM side-channels with watermarking to establish ownership of an IC and the proof required in terms of conflict.

In this chapter we discuss the methods in which we implemented the watermark design for different platforms and how they were detected using EM side-channels.

6.1 Watermark Design

Our work is based on the scenario that the owner/verifier of the design inserts a small circuit which we call the Watermark and has the knowledge of the watermark and tries to keep it hidden in the remaining design. In order to verify whether there is any illegal use of her Intellectual property, she tries to detect the watermark she has inserted. If she could detect the watermark and in case there is an illegal use of her IP she can prove her ownership.

The Watermarking concept is similar to the Trojan concept introduced in [25] except that it is used for a constructive purpose, which is to avoid IP fraud and counterfeiting. Since this involves providing IP rights, we can assume that the owner should insert a circuit that provides her identity. So the Watermark design mainly consists of:

1. A key/unique ID that is specific for the owner or even the particular chip (assuming the owner is given a set of IDs for her set of products)
2. Combinational function that can make use of key/ unique ID and some incoming messages or random challenges
3. Leakage generating circuit that depends on the value of the computed combinational function to help the owner/verifier to detect the inserted watermark

In this work, we focus on both Software Watermarks and Hardware Watermarks. In the case of Software Watermarks, we make use of platforms that use Assembly level language like a Microcontroller and a Smartcard (which is of contact type). Figure 6.1 shows the assembly level code implementing a combinational functions that makes use of the watermark key which here is 35 202 and a Leakage generator code.

```
;watermark1:
   subi r16,35
   subi r17, 202
   mul r16, r17
   mul r1, r0
   ;mul r5, r3

;leakageGenerator:
   SBRC r0, 7
   neg r1
   ;sts lc, r7
   SBRC r0, 7
   neg r1
```

Figure 6.1 **Combinational Logic and Leakage generator for Software Watermarking**

For the Hardware Watermarking, we make use of FPGAs. We can use of a 64-bit key or 128-bit key depending upon the algorithm in use. In this work, we made use of a 128-bit key as the Unique ID. The “Control FPGA” sends inputs or plaintexts 8-bits at a time. The combinational function makes use of 8-bit slice of the 128-bit key and the incoming 8-bit Plaintext and produces an output Y. The key slices keep changing for every new set of inputs. Figure 6.2 shows the HW Watermark design used. We make use of a set of LFSR or Ring oscillators as leakage generating circuits.

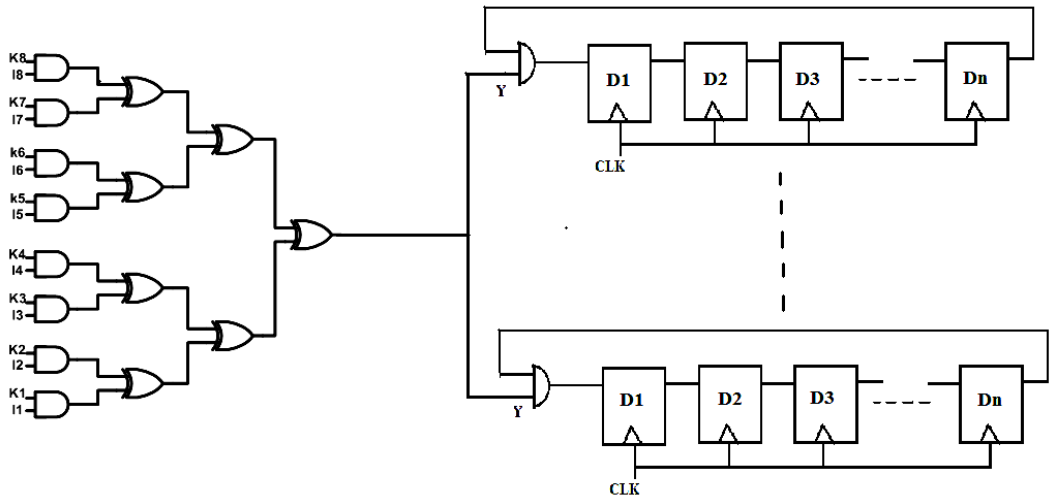


Figure 6.2 **Design for Hardware Watermarking**

For all the setups we use a DPO7104 Tektronix oscilloscope for capturing the waveforms. We use ETS Lindgren 7405 Near Field probes for the EM traces and Tektronix TDP 1000 Differential Probe for the power traces.

6.2 Software Watermarks

6.2.1 Microcontrollers

The first platform we have targeted is the Microcontrollers as we know that side-channel attacks have been made on them a number of times and it is a useful starting point due to their step by step functioning, which can help in figuring out the information from the raw traces. We have made use of an ATMEGA8 microcontroller which helps in creating a very light weight watermark consisting of only 4 to 5 assembly level instructions, which can easily be hidden in the rest of the operation the microcontroller performs.

We created a 2 PCB setup using the ATMEGA8 microcontroller in one of the PCBs and used the other PCB to supply the different challenges using a serial connection and powered them both with different power supplies. The 2 PCB setup also isolates the core, which here is the microcontroller, from outside sources of noise such as the fluctuations of power supply from a PC and capacitances in the power supply line. This setup was mainly created in order to provide a clean platform for power analysis which is done by placing a resistor in series with the power supply line. Figure 6.3 shows the microcontroller setup used in this work.

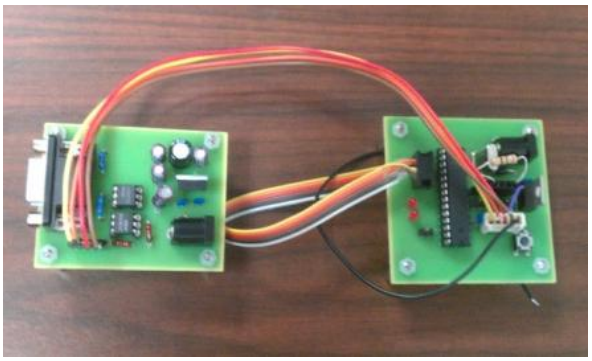


Figure 6.3 **Watermarking setup for microcontroller**

6.2.2 Smartcards

The next platform we have considered is the Smartcard platform, which is ubiquitous nowadays ranging from University cards, Laundry cards, Transportation cards and Credit cards. Hence this could potentially be a very relevant place where Side-channel watermarks could find its use. The platform is pretty similar to the microcontroller platform but finding a trigger signal is very hard.

We have used an ATMega 163 Fun card which has an 8-bit microcontroller based on the AVR Architecture. We have used SCM 3310 Smart card reader that has 8 contacts

and uses PC/SC API which works with Windows environment. In order to make it useful for both power and EM measurements, we have made a custom setup that uses a card slot that is connected via wires soldered across the reader contacts.

Figure 6.4 **Smart card setup for Watermarking. Left side setup shows EM probe on top of the smartcard while right side setup shows EM probe near the resistor where power measurements are taken**

Figure 6.4 shows the setup used for Smartcard watermarking. This setup is not made specifically for Power analysis and is allowed to be prone to noise.

6.3 Hardware Watermarks

We have targeted FPGAs for Hardware Watermarking where we use Hardware Description Language like VERILOG or VHDL. We have used 2 Spartan 3E FPGAs that contain a XC3S500E FPGA in 90nm 8-metal-layer process technology. The 1st one will be used as the “Core FPGA” which performs the watermark function and the remaining design. The other FPGA is used as a “Control FPGA” which sends different inputs through a serial connection.

This setup is to make the FPGA setup similar to the one used in SASEBO board which is a Standard Evaluation board for Side-channels Analysis. The FPGA logic core is powered by a standalone power supply or a normal 5V power supply although the stand alone supply from a adjustable voltage supply can have less disturbance and noise. The core power can be sensed by a 25mΩ current-sensing resistor with on-board jumpers. The EM radiations can be sensed using the EM probe placed on top of the FPGA chip. We have de-soldered the core power supply decoupling capacitors to achieve a higher resolution of the power traces. Figure 6.5 shows the Hardware Watermarking setup with EM and Power probes.

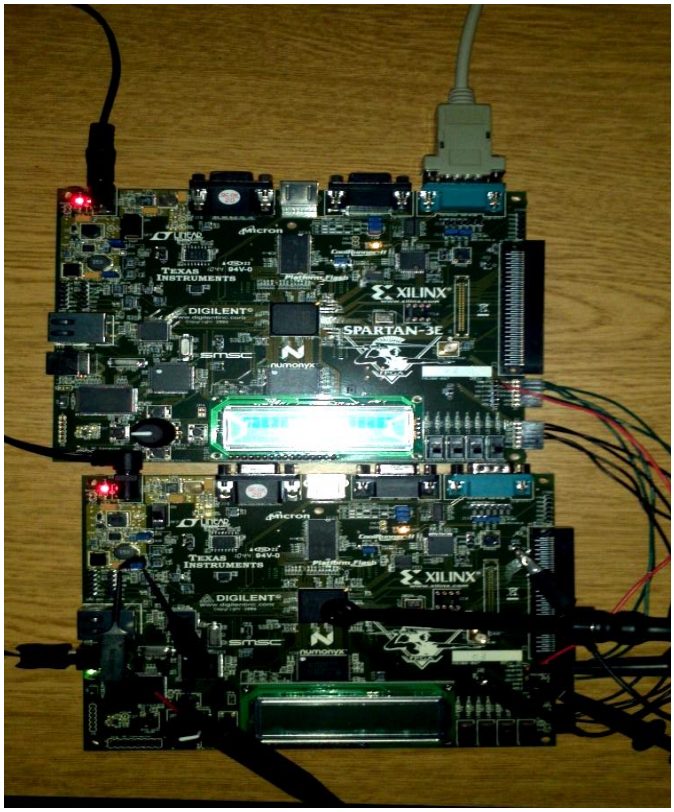


Figure 6.5 **Hardware Watermarking with the bottom FPGA as the "Core FPGA" and top FPGA as the "Control FPGA"**

6.4 Detection Of Watermarks

To detect the inserted Watermarks, we make use of the Differential Electromagnetic Analysis (DEMA) based on correlation. For this type of analysis, we need to take a number of measurements. Figure 6.4 shows the steps involved in the detection of the watermarks.

We follow the standard procedure for the Side-channel attacks whereby the assumption is that the owner/verifier can have access to the physical device and can capture many traces and also know either the inputs (Plaintexts) sent or the outputs (Cipher-texts) obtained from the system. Since the owner/ verifier will know what the watermark is, she can compute the output of the combinational function with the given inputs and the watermark key and store this value as the correct hypothesis. The owner/ verifier can then compute different hypotheses using different inputs and different key guesses. Only when the key is correct, the hypothesis can be correct. The owner/ verifier can then correlate all the hypotheses with the actual EM and power traces. She can prove that the watermark is inserted when she finds the high correlation peak that emerges for the correct key and inputs, as only the correct key will have the best prediction for the EM radiations and power consumption of the device.

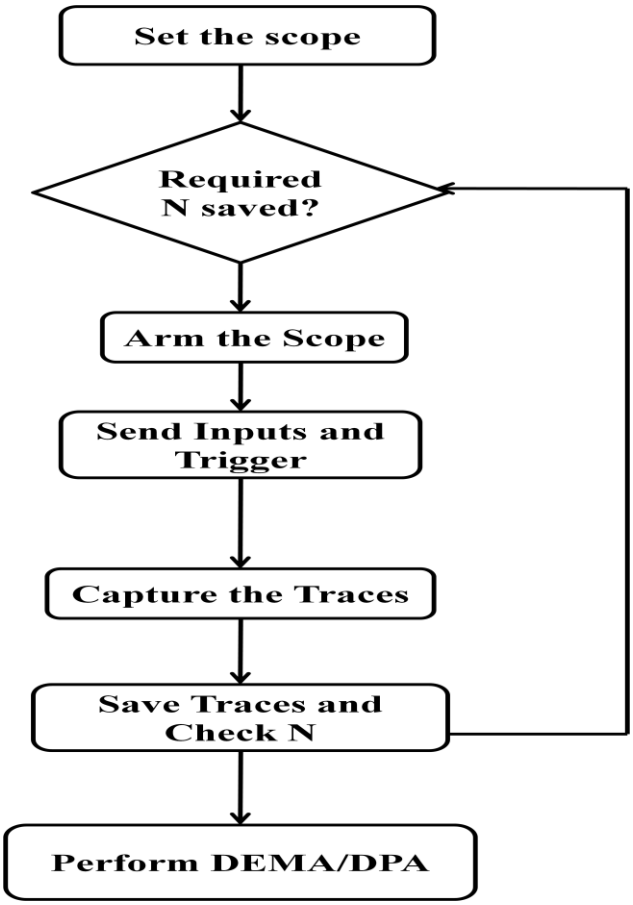


Figure 6.6 **Steps involved in Watermark Detection**

6.4.1. Software Watermarks:

For the microcontroller, we have a clean setup for power and therefore found that the watermark could be detected with as low as 1,000 traces and we have tried to obtain the EM traces just by placing it on top of the microcontroller without any depackaging. We know that the number of Side-channel papers based on power is way higher when compared to the papers from EM side-channel. The reason is that EM analysis requires more traces and efforts but we have found that with specific EM probes, we could detect the watermark in 5,000 traces when placed on top of the micro controller and just 1,000 traces when placed near the resistor if power setup is used. Figure 6.5, 6.6 and 6.7 show

the plot of Correlation for both Power and EM Side-channels where the different peaks reveal the different key bits. A positive peak means a 1 in the key and a negative peak means a 0 in the key. The design is such that different key bits are leaked at various time points, hence the number of peaks. The key bits are 11101001 for the 1st slice of key and 00100110 for the second slice and goes on for the different time periods. These are shown in boxes in the figures below. It could be seen from the power and EM traces that the Power traces have a very high correlation coefficient of 0.8 or 0.9 while the EM traces have a correlation coefficient of 0.3 or 0.4. We must take into the consideration that the setup is not affected by the noise in the case of power but EM signals are prone to noise from the other equipments and personnel in our lab. But we can still detect the key bits very clearly using the EM side-channel.

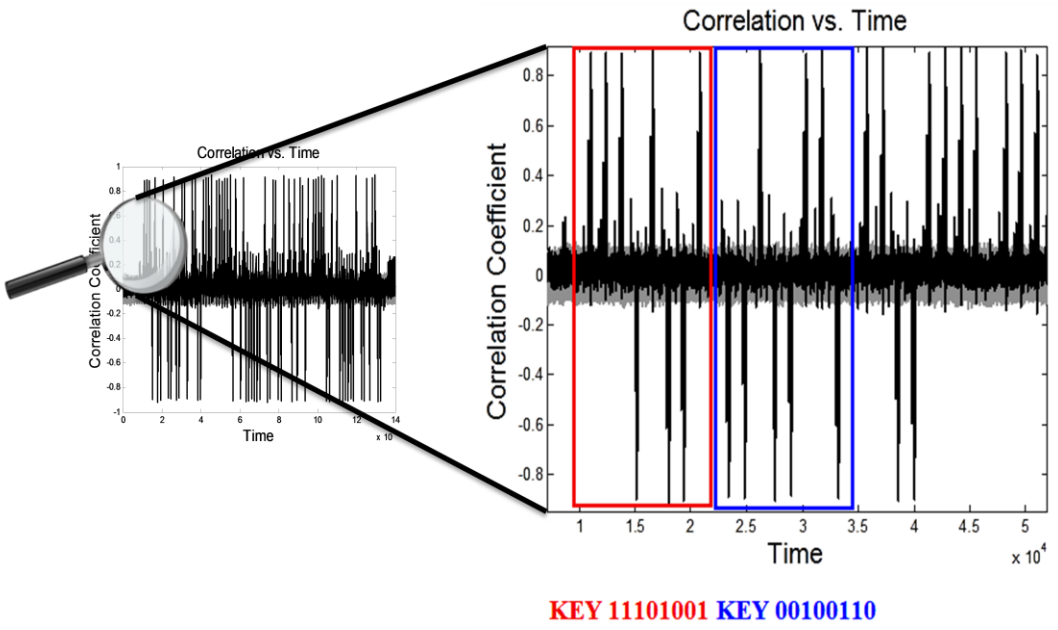


Figure 6.7 **Watermark in microcontroller detected using Power side-channels.**

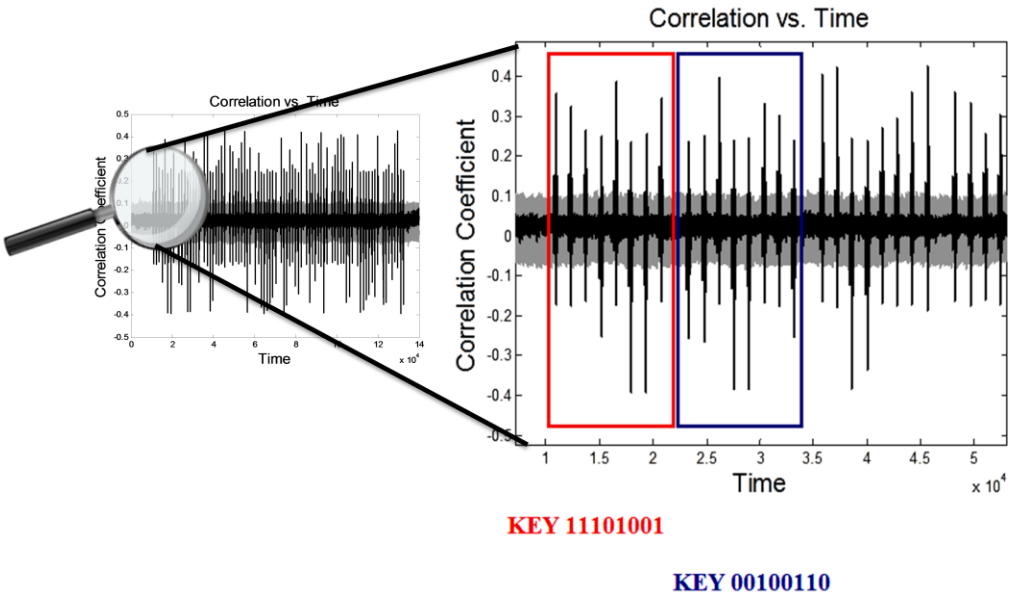


Figure 6.8 **Watermark in microcontroller detected using EM side-channels with EM probe near the resistor. Key bits 11101001 and 00100110 are visible from the plot.**

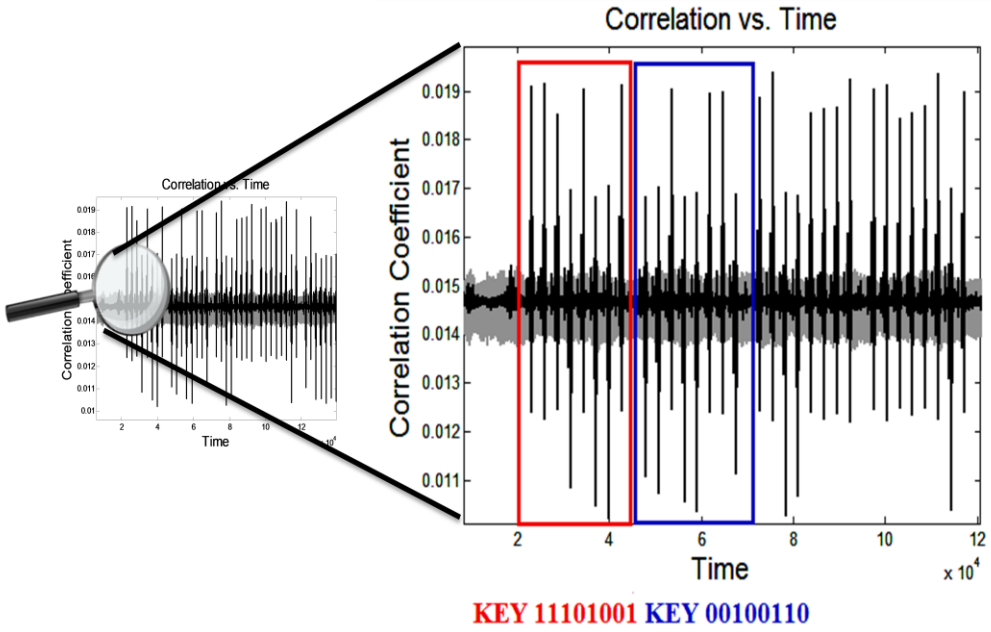
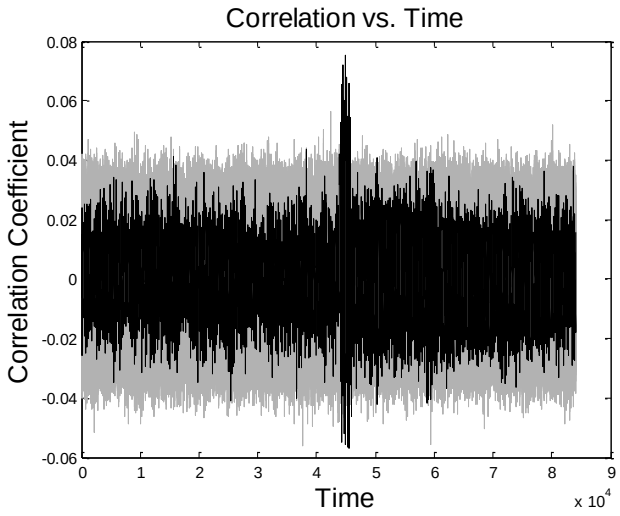


Figure 6.9 **Watermark detected using EM side-channels on top of the microcontroller. Key bits 11101001 and 00100110 are visible from the plot.**

In the case of Smartcards, the setup is made in such a way that although provisions have been made to accommodate the power and side-channels, both are prone to noise from outside sources. This is to mainly emphasize that the watermarks can be detected even if there are no custom boards and the only thing needed is the knowledge of the watermark, which is definitely possible since it is the owner and verifier who need to detect the watermark and they will design it. Figure 6.8 and Figure 6.9 show the Watermark detected at a specific time interval where the watermark key is being used by the assembly level instructions, and that only the correct key (watermark key/ unique ID) can give a high correlation peak. We set the key guess number 50 to be the correct key and hence a higher peak for the key guess when compared to the other wrong key guesses. This setup required 10,000 different inputs when power side-channels were used.



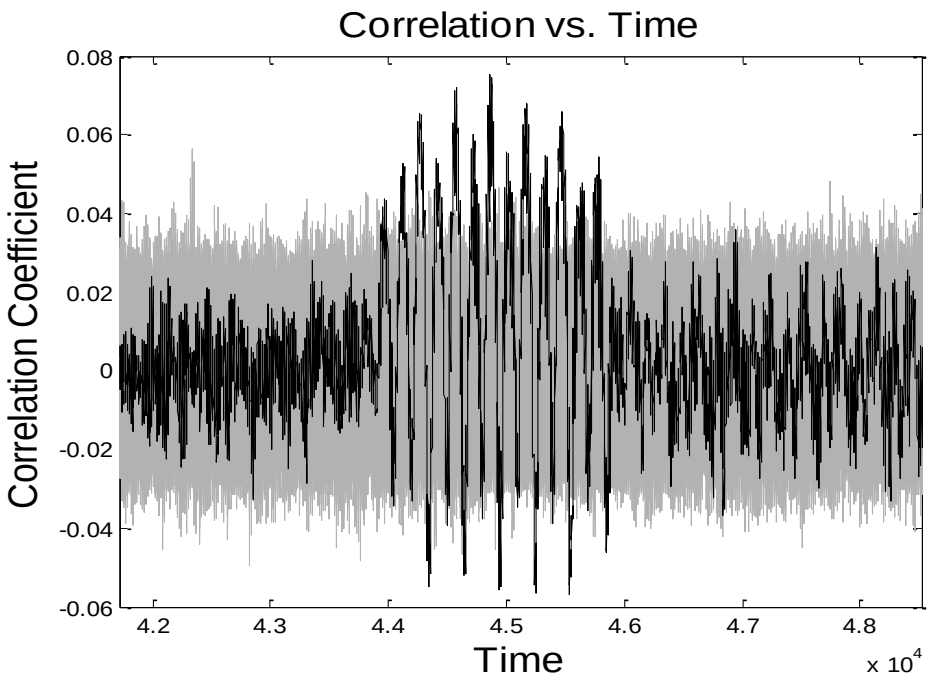


Figure 6.10 **Watermark in Smart card detected using Power side-channels. The top graph shows the watermark in the time domain. The bottom graph shows the zoomed version.**

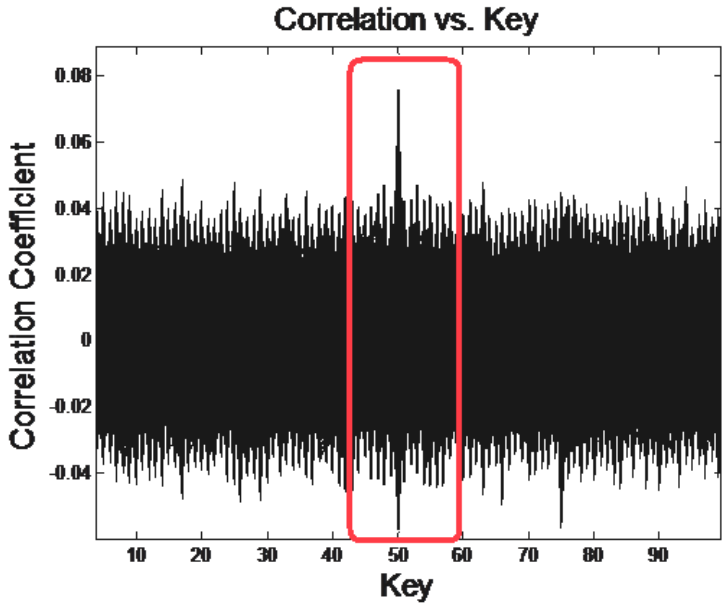


Figure 6.11 **Watermark in Smart card detected using power side-channel. The peak shows the correct key guess value is 50**

The same procedure is done for EM side-channel but only with 6,000 inputs. Figure 6.10 shows the watermark in smartcard detected using EM probes.

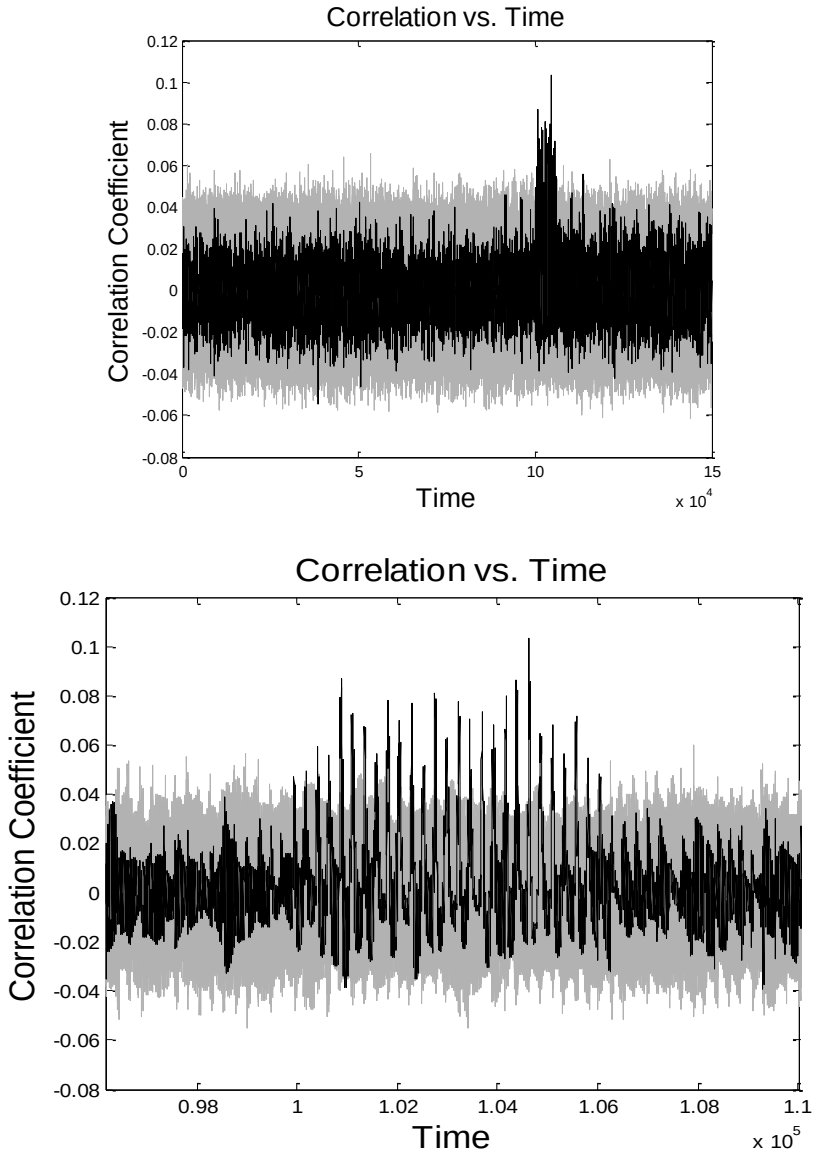


Figure 6.12 **Watermark in Smart card detected using EM side-channels. The top graph shows the watermark in the time domain. The bottom graph shows the zoomed version.**

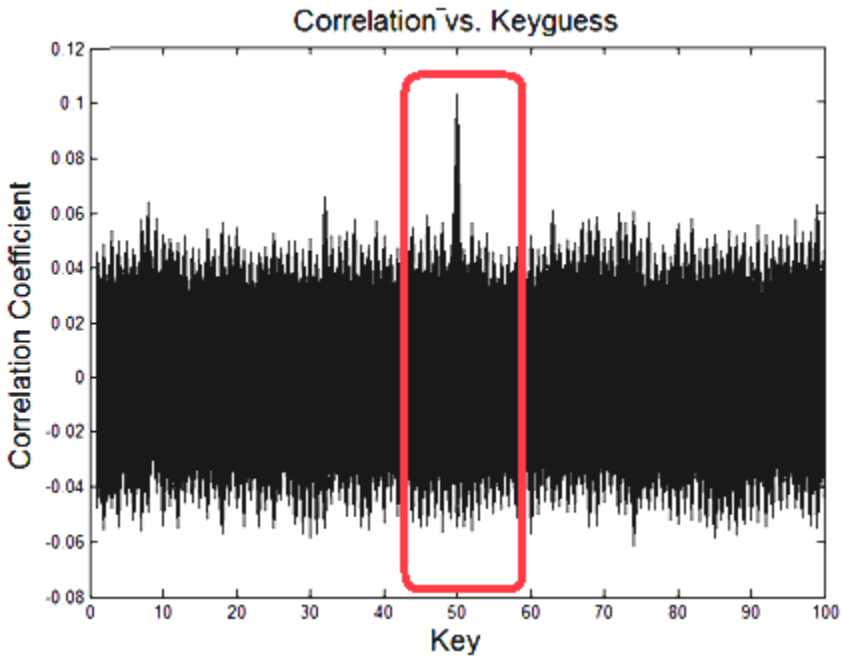


Figure 6.13 **Watermark in Smart card detected using EM side-channels. The peak shows the correct key is the 50th key guess.**

6.4.2. Hardware Watermarks

For hardware watermarks we have used FPGAs where we have implemented designs that will follow the watermark model explained in section 6.1. We have used the same combinational function as in [47], a 128-bit secret key which we have set to all FF, and for the leakage generating circuits we have used 10-bit shift registers which uses the combinational function output as its enable signal. We have random logic before and after the watermark so as to provide a relatively realistic model of how the design could be in commercial products. We provide a number of different inputs from the Control FPGA and a trigger signal through a serial connection and make the Core FPGA perform the operations in the design and watermark and capture the EM traces. We then perform correlation based analysis on the obtained traces to detect the watermark inserted. The

following figure shows the watermark with secret key 128'h FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF.

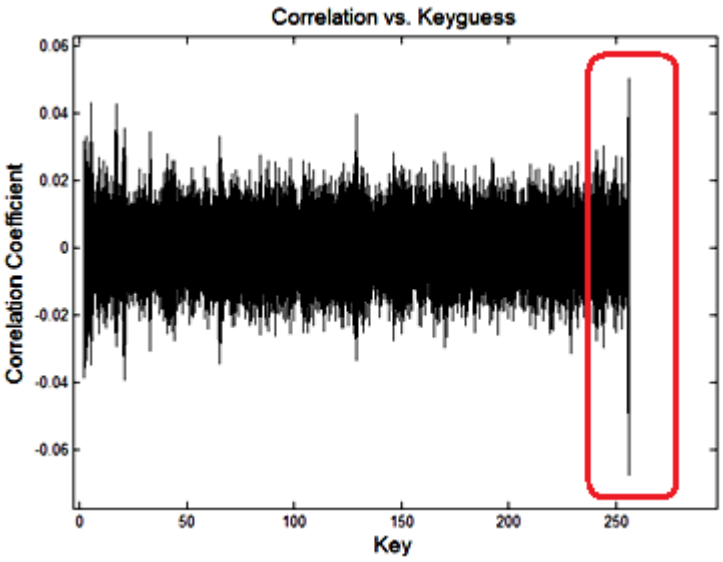


Figure 6.14 **Watermark in FPGA detected using EM side-channels. The peak shows the correct key is FF.**

We required 15,000 traces to find the watermark. Although the CC was around 0.07, we could clearly see the watermark with the unique ID of all FF. With more number of traces, the CC could increase and mainly the clarity of the peak against the other incorrect CC will be better. That is to say, the incorrect CC will go down as we increase the number of traces as unwanted noise will get averaged out.

In this implementation of hardware watermarking, we have connected the leakage generator output to an IO pin of the FPGA. We use a number of leakage generating shift registers but connect only one of it to an IO pin, primarily to give more signal strength and also to prevent our design from getting trimmed. When we do not use IO pins, the EM radiations created does not have enough strength for us to detect the watermarks. In this implementation we have made use of EM radiations from the FPGA and not the

waveforms at the IO pins as we have done in the Trojan implementation. So this is EM side channels detection for Watermarking. But we have connected the logic after the watermark to an IO pin so that the Xilinx ISE does not remove the underlying logic and in the process the whole design due to optimization. The Xilinx ISE checks for logic that does not drive or not getting driven and tries to minimize the logic. So in order for it not to "Trim" the designs and remove our whole design, we need some part of the design that is connected to IO pins or that drive logic that are connected to IO pins. All the other underlying logic and other instances of shift registers are prevented from getting optimized by the usage of "KEEP" property. But the Xilinx ISE did not allow all the shift registers to use KEEP property, as it then removed the logic saying there is no design that driving other logics.

This implementation trick gives more EM side channel and also helps the design to get synthesized and perform what we wanted it to do. Although, this implementation trick provides an added advantage for the EM radiations as IO pins in general provide more EM radiations. So though this implementation uses EM side-channel to prove the concept of watermarking, it makes use of the leverage caused by IO pins. We must concede that this procedure cannot be followed in real watermarking systems as this will not be small enough to hide, as the attacker can sense there is a high radiating signal and can lead to him checking the designs for watermarks. In this sense this cannot be termed a real Side channel usage, but this makes use of the EM side channels and can be termed a prototyping model for EM side channel based watermarking. In order to move ahead from such a prototyping to a "real" Side channel analysis, one must make sure the watermarking does not use advantage from the IO pin. Future work can target this by

making custom boards as in the case of ASICs and also maybe depackage the FPGA chip as this will increase the signal to noise ratio and prevent using the IO pins.

6.4.3. Comparison of Results

In this section we compare how the EM side-channel fares against Power side-channel. We consider three different platforms with varying levels of noise and interference.

The microcontroller setup is specifically made for power analysis which has two PCB boards with isolation to avoid power noise sources. We used the EM side-channels available from the top of the microcontroller as well as a place near the resistor. From Table 6.1 and Table 6.2, we can see that even though the setup is made for power, the watermark can be detected using EM side-channels with the same number of traces (1,000), if the EM probe is placed near the resistor. We can see that with 5,000 traces, EM side-channels can be used to find the watermark just by placing the probes on top of the microcontroller. This setup for EM side-channels does not even need decapsulation of the microcontroller (which drastically reduces the noise levels and provides better signal strength). We now see that the Coefficient correlation for finding the correct peak, which confirms the detection of watermark, is at just 0.3 to 0.4 for EM side-channels near the resistor, and 0.04 for EM side-channels on top of the microcontroller. This CC value will increase when de-capsulation of the microcontroller is possible. But it should be noted that the watermark can be detected with EM side-channels even with setups custom made for power analysis.

Table 6.1. Comparison of number of traces for Microcontroller

Side-channel	Number of Traces
EM near resistor	1000
EM on top uC	5000
Power	1000

Table 6.2. Comparison of CC for Microcontroller

Side-channel	Correlation Coefficient
EM near resistor	0.3 to 0.4
EM on top uC	0.04
Power	0.8 or 0.9

Table 6.3. Comparison of number of traces for Smartcard

Side-channel	Number of Traces
EM near resistor	10000
EM on top Smartcard	5000-6000
Power	10000

Table 6.4. Comparison of CC for Smartcard

Side-channel	Correlation Coefficient
EM near resistor	0.08 to 0.1
EM on top Smartcard	0.1 to 0.12
Power	0.08 to 0.1

The Smartcard setup is not specifically made for power analysis. There is no isolation of power supply, which will help the power analysis. We used the EM side-channels available from the top of the Smartcard as well as a place near the resistor. From Table 6.3 and Table 6.4 we can see that the watermark can be detected using EM side-channels with 5,000 traces, whereas the power side-channels require 10,000 traces. We can also see that the Correlation Coefficient for finding the correct peak, which confirms the detection of watermark, is almost equal and in fact 20% percent more than that obtained from power. This is a setup with noise to both EM and power where we can find that EM side-channels perform better in terms of effort required. Regarding the hardware watermarks, we could find the watermark with EM side-channels but we could not find the watermark using Power side channels with the existing setup where we just de soldered the coupling capacitances.

6.5 Robustness Analysis

In this section we will analyze how secure the inserted watermarks can be. This can be known by studying the possible attacks that the watermarks can be subjected to and mainly what their vulnerabilities are. As we know, the watermarks are assumed to be inserted by the owner of the chip or design under test. An attack in this scenario could be

- Removing the watermark from the original design
- Hiding the watermark so that it can't be detected by the owner/ verifier
- Creating new watermarks that can offset the original watermarks

The concept of watermarking has long been used for audio, video, and digital content. In such watermarks, there are four classes of attacks that could be carried out:

- ***Robustness Attacks***

In the case of image watermarking, Robustness attacks are those attacks that try or attempt to diminish or remove the presence of watermarks but try not to affect the original content. These can be accomplished by signal processing like filtering, resizing, and scanning, or analytic or algorithmic attacks like collusion attacks that combine different versions of the image to get a new image which inherently reduces the strength of watermark, making it difficult to be detected.

- ***Presentation Attacks***

This type of attack is where the watermarked image or digital content by itself is targeted and thus there is no need to hide or diminish the strength of the watermark. This will help in fooling an automated detector by giving wrong information if the automated detector is specifically looking for the watermark at a specific place in the content.

- ***Interpretation Attacks***

This type of attack is where the attacker can insert his own watermark and claim ownership. It will be difficult to find out whose watermark is original when both the watermarks have equal strength. This could again lead to argument, as there is more than one watermark in the same design

The attacker could also try to insert a watermark that can actually perform the inverse function of the original watermark thereby reducing its strength and hiding the original watermark.

- ***Legal Attacks***

This type of attack, as the name indicates, is mainly from the legal perspective and has nothing to do with the watermark design and insertion. These attacks mainly

depend on the laws of copyright and ownership, credibility of owner, and the attacker among other things.

In all the above mentioned attack options we can see that there has to be some knowledge of the presence, place, or type of the watermark for the attacker to succeed or it will very time consuming or impossible. So the first assumption is that the design will be kept secret by the owner and also small. Since this is the Identity of the IP that is in question, we assume that each company will be provided with different IDs/ keys space (so that many devices are watermarked). The main thing here is to keep the key/ unique ID secret. The other assumption is that there are no insider attacks. The watermarks will definitely be vulnerable to insider threats and can even be converted to become a Trojan as discussed in the previous chapter. This will lead to leaking out the secret ID which could be an IP breach, as this will provide the information to the attacker, who in this case is the illegal user who tries to copy the original design. If the attacks require more efforts and more time than that could be practically affordable, they will be called unbreakable in terms of cryptography.

Considering the above mentioned assumptions, we will now look into the different methods in which the inserted hardware and software watermarks could be attacked or rendered useless.

6.5.1. Reverse-engineering attack

This is similar to the robustness attack previously discussed where the aim is to find the watermark inserted in the design.

In the case of software watermarks, we have the watermark written in assembly level language. In order to reverse engineer, the attacker has to find the watermark once she has access to the assembly level code. Even if the attacker has access to it, finding a set of instructions that perform a specific operation is very difficult in assembly level, as this is not as straightforward as in higher level languages, such as C and Java. Also the watermark inserted is very small. Finding a watermark that is of less than 10 instructions will be practically very difficult when the original code is very huge. The amount of effort required will be so high that it is not worthwhile for the attacker to reverse engineer the assembly level watermark.

In the case of hardware, reverse engineering is a field in itself and is very difficult for an attacker, as it requires a large investment in effort and various special equipments are required. There are a series of steps to be followed to remove the chip layer by layer in order to access the design and wiring connections to know watermark location and operation. If an attacker is motivated enough and has resources to tear down the chip, almost any chip is vulnerable, but in general, the watermarks can be assumed to be safe against such reverse engineering attacks.

6.5.2. Side-channel attacks

We have made use of the EM side-channel analysis principle for a constructive purpose but the same can be used by an attacker for finding the watermark. But the problem for the attacker is that he does not know the secret key and leakage circuit which we have assumed to be secret and company specific. The attacker's aim here will be to remove the watermark once he finds it to avoid further problems from the verifier/ owner.

The attacker can find this by analyzing the exact clock cycles where the watermark operation takes place and can try to find what instructions they are and remove it in the case of software watermarks. For hardware, he can alter the codes only in the design phase or even change the watermark to become a Trojan so that it leaks unwanted information which he could claim as his watermark.

Though side-channel analysis proves to be a possible option for an attacker, again it will require considerable effort for the attacker if the key, leakage circuit, and combinational circuit remain secretive and mostly small so that the side-channel information is not provided to the attackers without special equipments to reveal them. When the combinational function and key space are large, it is practically impossible for any attacker to find the watermark without inside knowledge.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1. Conclusions

In this work, we have made use of side-channel analysis (mainly EM analysis) to attack FPGA with encryption algorithms, and also used it for Watermarking applications for microcontroller, smart card and FPGAs. We have used EM side-channels to make use of the advantage of not requiring the resistor to be placed in the power supply path, as this helps in the automation of watermarking applications for a manufacturer. The EM side-channels will require just an EM probe to be placed near the chip in use and does not require any modifications to the supply path or the design under test. This work required creation of setups for Attacks and Defenses using side-channel specific probes and hardware. We have made use of MATLAB for all the processing and supply of inputs in the case of microcontrollers and FPGA. We have also made use of C for the smartcard environment which uses Microsoft PC environment, WinScard libraries and functions. We have also made use of AVR studio for the codes in assembly level language for the Smartcards. This is the first time that EM analysis is used for watermarking and also hardware Trojans.

What we have found out is that EM side-channel analysis, like Power analysis, can be used to attack secret keys, insert, and detect watermarks in both hardware and software. Although EM analysis requires more effort in terms of signal processing and number of hours spent in capturing traces and analyzing, it is pretty effective in finding out the secretive information and protecting them. We found that in terms of hardware

with noise interference, EM side-channels can provide equivalent and sometimes better results when compared to power analysis, which was evident from the correlation co-efficient obtained from the detection of the watermarks in terms of smart card. Also, since this thesis was the first work in terms of actual hardware for our lab, all the required setup was developed and at this point provides setup for both smartcards and FPGAs.

7.2. Future Work

In this work we have made use of the near field EM side-channel information for both software and hardware watermarking and attacks. The main advantage of EM side-channel over Power side-channel is localization, where one can use a number of very tiny EM probes that can actually touch or monitor the contact pins in terms of hardware. Due to time and resources, we did not focus on utilizing this property of EM side-channel and this is a possible future direction in terms of the work carried out here. Also the work here increments and provides support for the work carried out by power side-channels and is the first work with EM side channels in the case of both microcontroller and contact based smartcards. A possible future direction would be to carry out the EM analysis on contactless smartcards and RFIDs where EM side-channels can play a big role.

This work concentrated on using only the side channel that is emanated from the design under test. Future works could possibly address the fact that the side channel information could be used for communication between two designs under test. Future direction could also try to study the usage of side channel information for 2 way communication and not just that is emanated. We could possibly insert side channel into

the design under test [49]. This could potentially lead to more security concerns. This would potentially lead to certain specifications to be followed while manufacturing hardware and implementing designs so as to allow (and reject) information coming into the designs.

In our group, we have been using passive analysis in all our work. We could possibly move into the direction where we could actively tamper with the designs, build equipments for fault injection type of attacks. This would require sophisticated equipments which could see through the layers of a chip. This would need a lot of effort and money to be invested, but would be necessary in order to specialize in all forms of security research.

The setup used in this work although being cheap and easy to construct, is very basic and mostly not noise tolerant. With better resources and more time, clean custom setups for EM side channel can be used in the future. A possible direction is to have a separate lab space exclusive for side-channel measurements. Usage of a faraday cage could help in mitigating noise from the environment. Multiple tiny probes could be used at different parts of the design under test to obtain more information.

BIBLIOGRAPHY

[1] W. van Eck. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security*, vol. 4, pp. 269–286, 1985.

[2] Paul Kocher. Timing Attacks on Implementations of Diffie Hellman, RSA, DSS, and other systems. Lecture Notes in Computer Science; Vol. 1109. *In the Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*. Pages: 104 - 113

[3] Paul Kocher, Joshua Jaffe, and Benjamin Jun- Differential Power Analysis.. August 1999. Crypto *'99. Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology.*

[4] J.-J. Quisquater and D. Samyde. Electromagnetic analysis (EMA): Measures and countermeasures for smart cards. In *Proceedings of e-Smart 2001*, Lectures Notes in Computer Science (LNCS), vol. 2140, pp. 200–210, Springer, 2001.

[5] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In C¸ . K. Koc¸, D. Naccache, and C. Paar editors, *Proceedings of CHES 2001*, Lecture Notes in Computer Science, vol. 2162, pp. 251–261, Springer,2001.

[6] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM sidechannel(s). In B. Kaliski, C¸ .K. Koc¸, and C. Paar editors, *Proceedings of CHES 2002*, Lecture Notes in Computer Science, vol. 2523, pp. 29–45, Springer, 2002.

[7] Eric Peeters, Francois-Xavier Standaert, Jean-Jacques Quisquater Power and Electromagnetic Analysis: Improved Model, Consequences and Comparisons. *Integration, the VLSI Journal, 2007*. Volume 40, Issue 1, January 2007, Pages 52-60 *Embedded Cryptographic Hardware*.

[8] Dakshi Agrawal Bruce Archambeault Josyula R. Rao, Pankaj Rohatgi . The EM Side–Channel(s):Attacks and Assessment Methodologies. *In Proceedings of CHES '02.*

[9] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, B. S. Kaliski Jr., C¸ . K. Koc¸, and C. Paar, Eds., 2002, vol. 2523 of *LNCS*, pp. 172–186, Springer-Verlag.

[10] D. Agrawal, J. R. Rao, and P. Rohatgi, "Multi-channel attacks," in *Proceedings of the 5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, C. Walter, C¸ . K. Koc¸, and C. Paar, Eds., 2003, vol. 2779 of *LNCS*, pp. 2–16, Springer-Verlag.

[11] C. D. Walter and S. Thompson, "Distinguishing exponent digits by observing modular subtractions," in *Proceedings of Topics in Cryptology - CT-RSA*, D. Naccache, Ed., 2001, vol. 2020 of *LNCS*, pp. 192–207, Springer-Verlag.

[12] V. Carlier, H. Chabanne, E. Dottax, and H. Pelletier, "Electromagnetic side-channels of an FPGA implementation of AES," *Cryptology ePrint Archive-2004*/145, 2004

[13] De Mulder, E.; Ors, S.B.; Preneel, B.; Verbauwhede, I." Differential Electromagnetic Attack on an FPGA Implementation of Elliptic Curve Cryptosystems**."** *Automation Congress, 2006.* WAC apos;06. World Volume , Issue , 24-26 July 2006 Page(s):1 – 6

[14] *T.* Ordas, M. Lisart, E. Sicard, P. Maurine, L. Torres "Near field mapping system to scan in time domain the magnetic emissions of integrated circuits", *In proc 18th International Workshop on Power and Timing Modeling Optimization and Simulation(PATMOS), Sept 2008*

[15] T. Ordas, A. Alaeldine, P. Maurine, M. Lisart, R. Perdriau, L. Torres, M. Ramdani "Evaluation of Countermeasures against Electromagnetic Analysis", *Electromagnetic Compatibility - EMC Europe, 2009 International Symposium, 11-12 June 2009 Page(s):1 - 4*

[16] Ali Alaeldine, Thomas Ordas, Richard Perdriau, Philippe Maurine, Mohamed Ramdani, Lionel Torres, M'hamed Drissi "Assessment of the Immunity of Unshielded Multi-Core Integrated Circuits to Near-Field Injection", *HAL: lirmm-00394411, version - 11 Jun 2009 (*Electromagnetic Compatibility, 2009 20th International Zurich Symposium *)*

[17] Eisenbarth, t., Kasper, t., Moradi, a., Paar, c., Salmasizadeh, m., and Shalmani, m. T. M. 2008. Physical cryptanalysis of keeloq code hopping applications. *Cryptology eprint archive, report 2008/058*

[18] Thomas Wollinger, Jorge Guajardo, Christof Paar. Security on FPGAs: State-of-the-Art Implementations and Attacks. August 2004. *Transactions on embedded computing systems (TECS),Volume 3 Issue 3.*

[19] Kaiyan Chen, Qiang Zhao, Peng Zhang, Gaoming Deng The Power of Electromagnetic Analysis on Embedded Cryptographic Ics. *The 2008 International Conference on Embedded Software and Systems Symposia (ICESS2008)*

[20] http://www.research.ibm.com/people/a/agrawal/publications/CryptoBytes2003.pdf 2009

[21] Tiri et al, *Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards*, ESSCIRC'02, p. 403

[22] Tiri et al, *Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation*, DATE'04, p. 246.

[23] Kris Tiri and Ingrid Verbauwhede. “A VLSI Design Flow for Secure Side-Channel Attack Resistant ICs”. *Proceedings of the conference on Design, Automation and Test in Europe - Volume 3*. Pages: 58 – 63

[24] Y. Jin, Y. Makris. “Hardware Trojan Detection Using Path Delay Fingerprint”. In *1st IEEE International Workshop on Hardware-Oriented Security and Trust* (HOST), pp. 51–57, 2008.

[25] L. Lin, M. Kasper, T. G¨uneysu, C. Paar, and W. Burleson. “Trojan Side-Channels: Lightweight
Hardware Trojans through Side-Channel Engineering”. In Cryptographic Hardware and Embedded Systems - CHES 2009, volume 5747 of LNCS, pages 382–395. Springer, 2009.

[26] G.T. Becker, M. Kasper, A. Moradi, and C. Paar. Side-channel based watermarks for integrated circuits. *In IEEE International Symposium on Hardware-Oriented Security and Trust* (HOST 2010), pages 30 –35, 2010

[27] Farinaz Koushanfar, Igor Markov. “Designing Chips that protect themselves”. Front End Topics. *Proceedings of the conference on Design Automation Conference* (DAC 2010).

[28] Jean-Francois Gallais, Johann Großschädl, Neil Hanley, Markus Kasper, Marcel Medwed, Francesco Regazzoni, Jörn-Marc Schmidt, Stefan Tillich, Marcin Wojcik - "Hardware Trojans for Inducing or Amplifying Side-Channel Leakage of Cryptographic Software" - *Trusted Systems. Second International Conference, INTRUST 2010*, Beijing, China, December 13th-15th, 2010.

[29] Georg T. Becker, Markus Kasper, Amir Moradi and Christof Paar. “Side-Channel based Watermarks for IP Protection”. COSADE 2010 - First International Workshop on Constructive Side-Channel Analysis and Secure Design.

[30] Christof Paar, Jan Pelzl. Understanding Cryptography. A Textbook for Students and Practitioners.

[31] Cryptography Research Inc. (Notes from One Day Side-channel Workshop)

[32] http://csrc.nist.gov/publications/nistpubs/800-12/800-12-html/chapter19.html

[33] www.poly.edu/csaw-embedded

[34] D. J. Wheeler, R.M Needham. “TEA, a Tiny Encryption Algorithm”

[35] A. Bogdanov , L. R. Knudsen , G. Le , C. Paar , A. Poschmann , M. J. B. Robshaw , Y. Seurin , C. Vikkelsoe. “PRESENT: An Ultra-Lightweight Block Cipher”. Lecture Notes in Computer Science 4727/2007 (2007)

[36] G. Becker, M. Kasper, A. Moradi and C. Paar, "Side-channel based watermarks for integrated circuits" IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2010

[37] G. Becker, W. Burleson and C. Paar, "Side-channel Watermarks for Embedded Software", 9th IEEE NEWCAS Conference (NEWCAS 2011), Bordeaux, France, June 2011.

[38] D. Agrawal, B. Archambeault, J. Rao, P. Rohatgi, "The EM side-channel(s): attacks and methodologies", In Proceedings of In Proc. Workshop on Cryptographic Hardware and Embedded Systems 2002, Aug. 2002

[39] F. Koushanfar, G. Qu, M. Potkonjak. "Intellectual property metering", Workshop on Information Hiding (IHW), pp. 87-102, LNCS vol. 2137, Springer-Verlag, April 2001

[40] J. Hamilton and S. Danicic. "An evaluation of static java byte code watermarking". In Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering and Computer Science 2010, pages 1 – 8, San Francisco, USA, October 2010.

[41] W. Zhu and C. Thomborson. "Algorithms to watermark software through register allocation". In Digital Rights Management. Technologies, Issues, Challenges and Systems, volume 3919 of Lecture Notes in Computer Science, pages 180–191. Springer Berlin / Heidelberg, 2006.

[42] C. S. Collberg, A. Huntwork, E. Carter, G. Townsend, and M. Stepp. "More on graph theoretic software watermarks: Implementation, analysis, and attacks". Inf. Softw. Technol., 51:56–67, January 2009.

[43] Cas interface user Manual. http://www.qboxsvn.com/duolabs/ManualeCas3_EN.pdf

[44] Wolfgang Rankl. Smartcard tables. http://www.wrankl.de/SCTables/SCTables.html

[45] Microsoft Smartcard Functions. http://msdn.microsoft.com/en-us/library/ms937004.aspx

[46] X. Wang, M. Tehranipoor, and J. Plusquellic, "Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions," Proc. IEEE Int'l Workshop Hardware-Oriented Security and Trust (HOST 08), IEEE Press, 2008, pp. 15-19.

[47] G. Becker, A. Lakshminarasimhan, L. Lin, S. Srivathsa, V. Suresh, and W. Burleson. "Implementing hardware trojans: Experiences from a hardware trojan challenge". To appear in International Conference on Computer Design (ICCD 2011), October 2011

[48] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, "Trustworthy hardware: Identifying and classifying hardware Trojans," IEEE Computer, vol. 43, no. 10, pp. 39–46, Oct. 2010.

[49] J.Sun, R.Bittner, and K. Eguro, “FPGA side-channel receivers,”In Proceedings of the 19th ACM/SIDGA International symposium of Field Programmable Gate Arrays (FPGA 2011)