Side Channel Analysis of a Java-based Contactless Smart Card

by

Edgar Mateos Santillan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Smart cards are widely used in different areas of modern life including identification, banking, and transportation cards. Some types of cards are able to store data and process information as well. A number of them can run cryptographic algorithms to enhance the security of their transactions and it is usually believed that the information and values stored in them are completely safe. However, this is generally not the case due to the threat of the side channel. Side channel analysis is the process of obtaining additional information from the internal activity of a physical device beyond that allowed by its specifications. There exist different techniques to attempt to obtain information from a cryptosystem using other ways than the normally permitted. This thesis presents a series of experiments intended to study the side channel from a particular type of smart card, known as Java Cards. This investigation uses the well-known technique, Correlation Analysis, and a new type of side channel attack called fast correlation in the frequency domain to study the side channel of Java Cards. This research presents a giant magnetoresistor (GMR) probe and for the first time, this type of sensor is used to investigate the side channel. A novel setup designed for studying the side channel of smart cards is described and two metrics used to evaluate the analysis results are presented. After testing the GMR probe and methodology on electronic devices executing the Advanced Encryption Standard (AES), such as 8-bit microcontrollers and 128-bit AES implementations on FPGAs, these techniques were applied to analyse two different models of Java Cards working in the contactless mode. The results show that successful attacks on a software implementation of AES running on both models of Java Cards are possible.

Acknowledgements

I want to thank my wife Vicky Chavez for her patience, and encouragement.

I thank my Supervisor Dr. Cathy Gebotys for her time, dedication, guidance and funding.

I thank the external Examiner Dr. Maire O'Neill and the members of my PhD committee, Dr. Gordon B. Agnew, Dr. William D. Bishop, and Dr. Alfred J Menezes, for their time, advice and great comments that help me improve this research work.

I thank Dr. David Nairn for giving me access to his signal generators.

I thank my mother Rosario Santillan and my siblings Guadalupe, Octavio, Oscar, Gloria, Graciela, Mireya, Arturo, Martha and especially my brother Juan Jose for their encouragement and advice.

I thank Wayne Loucks, Roger Sanderson, Brian White, Marcio Juliato, Jem Berkes, Farhad Haghighizadeh, Patrick Longa, Masoumeh Dadjou, Solmaz Ghaznavi, Amir Khatibzadeh, David Kenney, Noman Hai, and Zhao Li, for their technical discussions and friendship.

I thank all my friends including Adriana Cruz, Alberto Ramirez, Malena Cardoso, Isaias Ramirez, Ruri Takayanagi, Abel Lopez, Lulú Resendiz, Julio Noriega, Betty Aguilar, Sergio Torres, Lilia Cruz, Agustín Dominguez, Fernando Rivero, Fabiola Sosa, Jorge Vazquez, Nelly Carvajal, Luis Ricardez, Margot Tello, Manuel Tavara, Elena Ruiz, Brenda Ornelas, Lotar Lilge, and Laura Marti.

I thank Mexican taxpayers and CONACYT for sponsoring my studies.

This research was supported in part by CONACYT and grants from NSERC and OCE.

Dedication

To my beloved wife Vicky and my son Edgar

To the memory of my daughter Rosita (†2011)

Table of Contents

AUTHOR'S DECLARATION ii

Abstract iii

Acknowledgements iv

Dedication v

Table of Contents vi

List of Figures ix

List of Tables xiv

Chapter 1 Introduction 1

1.1 Contributions 4

1.2 Thesis Organization 4

Chapter 2 Introduction to Smart Cards 7

2.1 Classification of the Smart Cards 7

2.1.1 Hardware 8

2.1.2 Software 10

2.2 Types of Smart Cards 10

2.2.1 Contact Cards 11

2.2.2 Contactless Cards 12

2.2.3 Dual Interface 13

2.3 Communication Protocols on Contactless Smart Cards 15

2.3.1 Type A Protocol 15

2.3.2 Type B Protocol 17

2.4 Security in Smart Cards 18

2.4.1 AES 19

2.5 Summary 22

Chapter 3 Side Channel Analysis 23

3.1 Power Analysis 24

3.2 EM Analysis 26

3.3 Correlation Analysis 28

3.3.1 Correlation Analysis in the Time Domain 28

3.3.2 Correlation in the Frequency Domain 30

3.3.3 Proposed Fast Correlation in the Frequency Domain 31

3.4 Evaluating the Effectiveness of an Attack 32

3.5 Experimental Setup and Initial Preliminary Results 36

3.5.1 Analysis of Microcontroller AT89C51ED2 36

3.5.2 Analysis of Traces from the DPA book 43

3.6 Comparison to Previous Research and Summary 45

Chapter 4 The DPA Contest Version 2 47

4.1 DPA Contest Setup 48

4.2 DPA Contest Results 49

4.3 DPA Contest Public Database 53

4.4 Comparison to Previous Research and Summary 56

Chapter 5 Side Channel Analysis using Giant Magneto-Resistive Sensors 57

5.1 Introduction to GMR 57

5.2 Preliminary Results using Giant Magneto-Resistive Sensors 59

5.2.1 Correlation Analysis in the Time Domain 60

5.2.2 Correlation Analysis in the Frequency Domain 63

5.3 Comparison to Previous Research and Summary 66

Chapter 6 Experimental Setup for Java Card Analysis 69

6.1 Card Reader Modifications 69

6.1.1 The Axis Table. 69

6.1.2 Power Supplies 72

6.1.3 Signal Generator 73

6.2 Triggering the Oscilloscope 73

6.2.1 Using the Pause at the Start of a Command Transmission 73

6.2.2 Using a Customized Circuit for Triggering 74

6.3 Oscilloscope Acquisition Modes 74

6.4 Applet Used for Analysis 75

6.4.1 Timing and Resources Constraints 76

6.4.2 Processing Times 77

6.5 Trace Acquisition 81
6.5.1 The Faraday Cage 81
6.6 Comparison to Previous Research and Summary 83
Chapter 7 Empirical Analysis of the Java Smart Cards 85
7.1 Java Card JCOP30 with Modifications 85
7.1.1 Time Domain Analysis 86
7.1.2 Frequency Domain Analysis 91
7.2 Java Card JCOP30 without Modifications 95
7.2.1 Time Domain Analysis 95
7.2.2 Frequency Domain Analysis 100
7.2.3 Measuring the Effectiveness of the Analysis in the Time and Frequency Domain 105
7.3 Java Card JCOP41 Without Modifications 106
7.3.1 JCOP41 Time 107
7.3.2 Frequency Domain Analysis Using the JCOP41 111
7.3.3 Measuring the Effectiveness of the Analysis in the Time and Frequency Domain 115
7.4 Comparison to Previous Research and Summary 115
Chapter 8 Discussions and Conclusions 119
8.1 Summary of Contributions 126
8.2 Future Work 126
Appendix 129
A.1 Java Code that Implements Applet Test 2 (S-box 21 Iterations) 129
A.2 ASK Decoder 132
References 133

List of Figures

Figure 2.1: General classification of cards 7

Figure 2.2: Smart card dimensions according to ISO/IEC 7816-1 and ISO/IEC 7816-2. 9

Figure 2.3: Smart Contacts of the Smart Card Module according ISO/IEC 7816-2......... 11

Figure 2.4: Typical architecture of contact microprocessor card..................................... 12

Figure 2.5: Typical architecture of a contactless card. It includes a microcontroller and a RF interface in the same chip.. 13

Figure 2.6: Microprocessor and antenna in a JCOP30... 14

Figure 2.7: Modulation ASK 100% with Modified Miller coding. 16

Figure 2.8: Load modulation with subcarrier using Manchester coding. 16

Figure 2.9: Modulation ASK 10% amplitude using NRZ-L coding................................. 17

Figure 2.10: Load Modulation with subcarrier, using BPSK NRZL coding. 18

Figure 2.11: Pseudocode for the AES cipher.. 19

Figure 2.12: ShiftRows( ) cyclically shifts the last three rows in the State. 20

Figure 2.13: Pseudo code for Key expansion. 21

Figure 3.1: Correlations in the frequency domain for the correct key guess using different system clocks. 37

Figure 3.2: Correlation analysis in the frequency domain for 2,048 EM traces using a system clock of 1 MHz. 38

Figure 3.3: Correlation analysis in the frequency domain for 55 EM traces using a system clock of 1 MHz. 39

Figure 3.4: Accuracy results after processing EM traces from our microcontroller......... 40

Figure 3.5: Computation time required to process a given number of traces using different correlation methods.. 41

Figure 3.6: Correlation analysis for one key byte using misaligned traces....................... 42

Figure 3.7: Comparison between the estimated effectiveness and the accuracy for different numbers of traces using the fast correlation in the frequency domain. 43

Figure 3.8: Correlation frequency analysis using the traces from the DPA book WS3.... 44

Figure 4.1 Results from the DPACv2 with the global success rate entropy for all the subkeys bytes 51
Figure 4.2 Results from the DPACv2 with the partial success rate for all the subkeys bytes 52
Figure 4.3 Results from the DPACv2 with the partial guessing entropy for all the subkeys bytes 52
Figure 4.4: Correlation frequency analysis using the traces from the DPA contest public database. 53
Figure 4.5: Processing time required to compute correlation analysis using traces from the DPA contest public database. 54
Figure 4.6: Accuracy results using the fast correlation in the frequency domain using power traces from the DPA contest public database. 55
Figure 4.7: Comparison between the estimated effectiveness and the accuracy for different numbers of traces from the DPA contest using fast correlation in the frequency domain. 56
Figure 5.1: Picture of the evaluation board used showing the position of the GMR probe (left). Close up of the GMR probe (right) 60
Figure 5.2: Correlation matrix in the time domain for 2,048 traces using EM probe and a sampling frequency of 500 MS/s. 61
Figure 5.3: Correlation matrix in the time domain for 2,048 traces using the GMR probe and a sampling frequency of 500 MS/s. 62
Figure 5.4: Correlation matrix in the time domain for 2,048 traces using EM probe and a sampling frequency of 50 MS/s. The analysis returns a wrong key. 62
Figure 5.5: Correlation matrix in the time domain for 2,048 traces using the GMR probe and a sampling frequency of 50 MS/s. 63
Figure 5.6: Correlation matrix in the frequency domain for 2,048 traces using EM probe and a sampling frequency of 500 MS/s. 64
Figure 5.7: Correlation matrix in the frequency domain for 2,048 traces using GMR probe and a sampling frequency of 500 MS/s. 65

Figure 5.8: Correlation matrix in the frequency domain for 2,048 traces using EM probe and a sampling frequency of 50 MS/s. 65
Figure 5.9: Correlation matrix in the frequency domain for 2,048 traces using GMR probe and a sampling frequency of 50 MS/s. 66
Figure 6.1: CardMan 5121, dual interface card reader 70
Figure 6.2: Layout of the axis table developed. 71
Figure 6.3: Axis table developed to place the smart card and EM sensor. 72
Figure 6.4: End of command from the card reader to the smart card JCOP41 for different plaintexts 79
Figure 6.5: Start of response from the smart card JCOP41 to the card reader 80
Figure 6.6: Segment of the setup used for analyzing the smart cards. 82
Figure 7.1: Representation of the correlation analysis matrix in the time domain using 39,936 traces. 87
Figure 7.2: Projection from the correlation analysis matrix in the time domain using 39,936 traces. 88
Figure 7.3: Projection of the correlation analysis matrix in the time domain using 39,936 traces showing the maximum correlations for each small part of the key guess in the interval 1,498.5 µs to 1,499.1 µs. 88
Figure 7.4: Hamming weight of the acquisition traces at the maximum correlation. On top the interval 1,498.5 µs to 1,499.1 µs and in the bottom a zoom to the interval 1,489.5 µs to 1,498.8 µs. 90
Figure 7.5: Representation of the correlation analysis matrix in the frequency domain using 39,936 traces. 92
Figure 7.6: Projection from the correlation analysis matrix in the frequency domain using 39,936 traces. 93
Figure 7.7: Projection of the correlation analysis matrix in the frequency domain using 39,936 traces showing the maximum correlations for each small part of the key guess in the interval 1,498.5 µs to 1,499.1 µs. 93
Figure 7.8: Hamming weight analysis in the frequency domain. 94

Figure 7.9: 3-D representation of the correlation analysis matrix in the time domain for the JCOP30 smartcard without modifications. 96

Figure 7.10: Projection of the correlation matrix for the time domain analysis using 5,888 traces showing the correlation values vs. time. 97

Figure 7.11: Projection of the correlation matrix for the time domain analysis using 5,888 traces. 98

Figure 7.12: Average of 5,888 traces according to its Hamming weight using the power model for the correct key guess and a sampling rate of 500 MS/s. 100

Figure 7.13: 3 D representation of the correlation matrix in the frequency domain using 5,888 EM traces. 101

Figure 7.14: 2-D projection from the max correlation per frequency after using 5,888 EM traces. 102

Figure 7.15: Projection of the Max correlations over all frequencies for the JCOP30 using 5,888 traces..102

Figure 7.16: Power spectrum averages according to their Hamming weights...............104

Figure 7.17: Comparison of 3 types of correlation analysis using accuracy as metric linear scale (left) and semi-log scale (right)..106

Figure 7.18: Representation of the correlation matrix in the time domain using 5,888 traces. 107

Figure 7.19: Projection from the correlation analysis matrix in the time domain showing correlation vs. time, using 5,888 traces..108

Figure 7.20: Projection of the correlation matrix in the time domain showing correlation vs. key guess using 5,888 traces. 109

Figure 7.21 Averages of the EM traces according to the Hamming weight of the correct key guess power model. 110

Figure 7.22: Representation of the correlation matrix in the frequency domain using 5,888 traces. 111

Figure 7.23: Projection of the correlation matrix after using 5,888 traces.....................112

Figure 7.24: Projection of the correlation analysis matrix for the JCOP41 using correlation analysis in the frequency domain and 5,888 power spectrum traces. 113

Figure 7.25: Average Hamming weight using the hypothetical power model of the correct key..114
Figure 7.26: Contrast between time and frequency domain analysis using accuracy as the metric (JCOP41). 115

List of Tables

Table 2.1: Sections of the ISO/IEC 7816 standard. 8

Table 2.2: Memory resources in different smart card models. 14

Table 2.3: S-box: substitution values for the byte *xy* in hexadecimal format. 20

Table 4.1 Results from the DPACv2 50

Table 5.1: Maximum correlation for the time domain analysis 61

Table 5.2: Maximum correlation values for the frequency domain analysis 64

Table 6.1: Processing time for the small testing applet 80

Table 7.1: Distribution of the 39,936 traces used according to their Hamming weight ... 89

Table 7.2: Test results for other keys used JCOP30 98

Table 7.3: Distribution of the 5,888 traces used according to their Hamming weight 99

Table 7.4: Test results for other keys used JCOP30 103

Table 7.5: Test results for other keys used JCOP41 113

Table 7.6: Summary of results for the correlation analysis presented. 117

Table 8.1: Characteristics of the analysis studies to the Java Card and
the DESFire Card. 124

Chapter 1

Introduction

Smart cards are used in many areas of modern life. They are employed in credit, debit, identification, and transport cards. They can store data and some types of cards are able to process information as well. They are used in a wide range of applications such as banking, transportation, and identification. In transportation applications, contactless cards are typically used, whereas in banking applications, contact, contactless or dual interface cards may be used. Some cards run cryptographic algorithms to enhance the security of their transactions and it is generally believed that the information and values stored in them are completely safe. However, this is generally not the case due to the threat of the side channel. Side channel analysis is the process of obtaining additional information from the internal activity of a physical device beyond that allowed by its specifications. There exist different techniques to attempt to obtain information from a cryptosystem using other ways than the normally permitted. Side channels can be imperceptible like in the case where EM emissions from a device are captured, the power consumption of equipment is monitored or the acoustic signals from a mechanical gadget are recorded (Rohatgi, 2006). Hence, if the electromagnetic (EM) signals emitted by the cards during cryptographic computations leak information related to the data processed, an attacker might be able to take advantage of that information and reconstruct the secret key. This work studies the security of one type of smart card called Java Cards, particularly the models JCOP30 and JCOP41.

The use of EM signals to retrieve information from electronic devices is nothing new. During World War II, the USA Army and Navy based their secure communications on an encrypting device call SIGTOT; nevertheless, the content of the message (or plaintext) could be determined examining the EM signals emitted by the machine (National Security Agency and Central Security Service, 1972). The U.S. National Security Agency in 1966, warned intelligence officers about possible information leakage through the use of the Cathode Ray Tube (CRT) on computers and console displays. The problem was that the CRT radiated strong signals and it was possible to recover in real time, fully legible copies of the information displayed on the screen from hundreds of meters away (Boak, 1973). More recently, some attacks on contact-based smart cards have been reported in the literature (Kocher et al. 1999), (Messerges et al. 2002), (Vermoen et al. 2007). Some attacks on contactless smart cards have been published in (Nohl and Plotz, 2007), (Koning et al. 2008), (Oswald and Paar, 2011) and they have motivated upgrades in the baseline of security. Card makers discontinued some models and replaced them by others certified in security labs (Mifare, 2008), (Mifare, 2011). These attacks have occurred to Mifare and DESFire specifications but not to Java Cards, until now. Attacking Java-based contactless cards present several challenges. They are powered by the card reader through an inducted EM field that helps to hide the tiny emanations from the IC of the card associated with cryptographic operations.

Smart cards are so important that their security is continually being tested. Power analysis and electromagnetic (EM) power analysis are strong side channel methodologies used to analyze hardware and software implementations of cryptographic algorithms. First, differential power analysis (DPA) was presented using smart cards (Kocher et al. 1999). From there a number of attacks have been reported using power analysis (Messerges et al. 2002), (Vermoen et al. 2007) and EM analysis (Quisquater and Samyde, 2001), (Gandolfi et al. 2001), (Agrawal et al. 2002), (Nohl and Plotz, 2007). There are cases where the authors have extracted the chip from the card to better study the EM since the signal to noise ratio (SNR) was too small with the card intact. Hence, they were unable to obtain any information (Carluccio et al. 2005). In other works, the researchers produce their own card readers and circuits that can even emulate the smart cards (Kasper et al. 2007). Some researchers have used power analysis to study Java Cards like in (Vermoen et al. 2007) where they develop their own card reader, measured the power consumption using a special system call Inspector (Riscure, 2011) and created a library with all possible instructions with their respective power traces. In this case, the analysis of cryptographic algorithms like DES or RSA was out of their research scope. In other research (Sterckx, 2009), the author use power analysis to study the implementation of anonymous credentials on Java Card smart cards. They found that the unstable internal clock made the traces strongly misaligned and impractical to attack. Another research (Berkes, 2008), does timing analysis on the contactless communication of Java Cards; however, EM side-channel analysis such as differential EM analysis (DEMA) was beyond their research scope.

In some studies, authors have used reverse engineering to detect security weaknesses and attack them (Nohl and Plotz, 2007), (Koning et al. 2008). In one case, they removed the chip from the card, polished the chip, and took pictures of all layers from the circuit. Using image recognition software, they recovered the electronic diagram of the card. After analyzing the diagram they found that, the cryptocore consisted of a 16-bit random number generator that uses a constant initial condition and a 48-bit linear feedback shift register (LFSR) (Nohl and Plotz, 2007). In 2011, researchers used an analog demodulator to enhance the EM signals from the analyzed smart cards (Kasper et al. 2011). In (Oswald and Paar, 2011) the authors go farther by successfully attacking the hardware implementation of 3DES from the DESFire3ICD40 using an analog demodulator and EM correlation analysis in the time and frequency domain.

One of the goals of this thesis is to study the security of Java Cards. For this purpose, a number of experiments are developed and analyses are performed searching for a possible side channel. Some of the experiments include the use of two models of Java Cards, JCOP30 and JCOP41, running a Java software implementation of AES. Unlike previous research where DESFire (Oswald and Paar, 2011) and Mifare Classic (Nohl and Plotz, 2007), (Koning et al. 2008) specifications have been attacked, Java Cards contain garbage collection services that the programmer cannot disable and these services automatically interrupt the program execution. Additionally, the overhead associated with the Java virtual machine makes the execution time extremely large compared with other types of cards or hardware implementations and crypto accelerators. Longer executing times increase the difficulties of searching for a possible side channel. This research uses the properties of the giant magnetoresistor (GMR) probe to acquire EM signals without the use of an additional analog demodulator or filter. All the signals studied use correlation analysis in the time and frequency domains.

1.1 Contributions

This thesis proposes a new experimental setup, analysis and set of metrics for studying the side channel from JCOP30 and JCOP41 Java Cards. First, a new type of side channel attack called fast correlation in the frequency domain is introduced and two metrics are detailed to evaluate the effectiveness of the side channel analysis. This type of analysis and the metrics are tested over different platforms and successfully recover the secret key on 8-bit and 128-bit systems. This attack based on fast correlation in the frequency domain was submitted to an international DPA competition (DPACv2), and resulted in the fastest attack among all entries. Among the fastest entries, it had the best success rate for 20,000 traces. For the first time a giant magnetoresistor (GMR) sensor was applied to acquire the side channel. These sensors are based on the magnetoresistance quantum mechanical effect discovered in the late 1980s to detect small magnetic fields. This probe (Mateos and Gebotys, 2011), successfully recovered the correct small part of the key from a microcontroller based system. Finally, a novel setup for Java Card analysis is described. It included the fabrication of a coordinate's analysis table and the design of an electronic circuit able to trigger the oscilloscope after a pre-programmed sequence of instructions are sent from the computer to a commercial card reader (Chapter 6). Finally, using EM emissions acquired with the GMR and the proposed correlation analyses, the correct small part of the key was recovered from the unmodified Java Cards, JCOP30 and JCOP41, running a software implementation of AES (Chapter 7).

1.2 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 presents a brief introduction to smart cards describing the processor cards and their characteristics. It covers contact cards, contactless cards, and dual interface cards like the Java Cards used for analysis later in this research. Additionally, it briefly describes the communication type A and type B protocols and the Advanced Encryption Standard (AES).

An introduction to side channel analysis describing the principles for power analysis and electromagnetic (EM) power analysis is presented in Chapter 3. Three variations of correlation analysis are explained, specifically correlation analysis in the time domain, correlation analysis in the frequency domain and fast correlation in the frequency domain. Then, two new metrics *Accuracy* and *Estimation* are presented. These metrics are used in this research to evaluate the effectiveness of side channel analysis. At the end of the Chapter 3, some preliminary results are reported using a microcontroller base system that is running a software implementation of AES. The EM traces used in the analysis were acquired using an inductive probe. Some of the experiments described include EM analysis using different clock frequencies, random misalignments and comparing the results of the three types of correlation analysis (time domain, frequency domain and fast correlation) using accuracy as a metric. Also previously published power traces from the DPA book WS3 (Mangard et al. 2010b) were analyzed with the new proposed correlation approaches.

An overview of the DPA Contest Version 2 (DPACv2) is reported in Chapter 4. The characteristics of the contest and the criteria used to compare the attack submissions are explained. Then, a brief report of the results of the DPACv2 and the analysis of the power traces from the DPACv2 public database are presented.

A new type of EM probe is described in Chapter 5, based upon a giant magnetoresistor GMR sensor. The performance from this probe is compared with a commercial inductive probe. The evaluations include analysis in the time and frequency domain with different sampling rates, (500 MS/s, 250 MS/s, 125 MS/s and 50 MS/s) when a software implementation of AES is running in a microcontroller based system.

The experimental setup designed to study the Java Cards used in this research is described in Chapter 6. First, a few modifications that were made to a commercial dual interface card reader to improve the quality of EM acquisitions are explained. Then, the functionality of an electronic circuit designed to demodulate the command from the card reader to trigger the digital oscilloscope used for capturing the EM traces is described. Then, two Java applets used for testing the security of the Java Cards are described, followed by an overview of the way the traces are acquired including one experiment where the IC from the card is removed and placed inside an enclosure to mimic a “Faraday cage”.

A summary with the most representative results from this research is presented in Chapter 7. First, the results of an experiment where the IC was removed from the JCOP30 card are shown, followed by experiments using the JCOP30 and JCOP41 cards. In this case, the cards used had no physical modifications. All the EM traces were acquired using a GMR probe and analysed using correlation analysis in the time domain and in the frequency domain. The analysis in the frequency domain was successful in retrieving the secret small part of the key in the three experiments presented while correlation analysis in the time domain was successful only for the JCOP30. Finally, the conclusions of this thesis with a summary of the contributions and future work are presented in Chapter 8.

Chapter 2

Introduction to Smart Cards

Smart cards are one type of embedded system that has become quite common in our daily lives. Given the multiple varieties of smart cards, this chapter focuses on the processor cards and describes their characteristics in contact cards, contactless cards and dual interface cards. Then, it outlines the physical characteristics, internal organization, and communication protocols for the type A and type B cards (contactless and dual interface cards, respectively) followed by a short description of the Advanced Encryption Standard (AES) algorithm.

2.1 Classification of the Smart Cards

Smart cards have progressed from basic memory cards in the early 1970s to microprocessor cards equipped with cryptographic processors, security sensors and USB 2.0 low speed contact interfaces (NXP, 2008). They have diversified from phone cards to credit/debit cards, transport cards, identification cards, electronic passports, etc. Even though the applications of smart cards are diverse, there are similar structures and patterns in the cards. It is possible to classify all types of cards according to Figure 2.1 (Rankl, 2007).

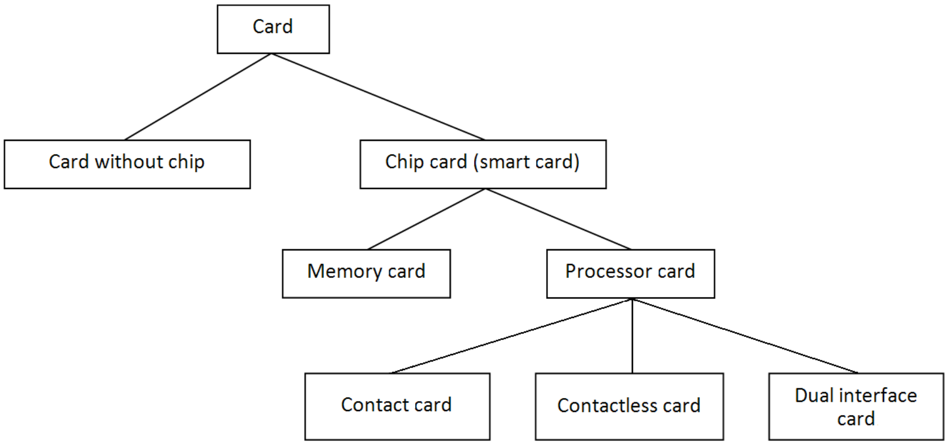


Figure 2.1: General classification of cards (Rankl, 2007).

The first division is between the cards without a chip and the cards with a chip, also known as smart cards. Among the smart cards there are memory cards, only able to store data, and microprocessor (processor) cards, able to store and process information. The microprocessor cards are divided into contact, contactless and dual interface cards. Dual interface cards are both contact and contactless cards. This research only covers aspects related to the microprocessor cards and focuses on the aspects of the dual interface card working in contactless mode.

2.1.1 Hardware

Each chip manufacturer typically includes specific features in their cards; however, the most basic characteristics are in the ISO/IEC 7816 standard. This standard specifies physical characteristics, dimensions and locations of the contacts, the electrical interface and transmission protocols, among other characteristics. This section refers to a few parts of the standard. Table 2.1 shows the different sections that this standard contains.

Table 2.1: Sections of the ISO/IEC 7816 standard (ISO/IEC, 2011).

Standard	Last modification	Subject
ISO/IEC 7816-1	1998	Physical characteristics
ISO/IEC 7816-2	2007	Cards with contacts -- Dimensions and location of the contacts
ISO/IEC 7816-3	2006	Cards with contacts Electrical interface and transmission protocols
ISO/IEC 7816-4	2005	Organization, security and commands for interchange
ISO/IEC 7816-5	2004	Registration of application providers
ISO/IEC 7816-6	2004	Interindustry data elements for interchange
ISO/IEC 7816-7	1999	Interindustry commands for Structured Card Query Language (SCQL)
ISO/IEC 7816-8	2004	Commands for security operations
ISO/IEC 7816-9	2004	Commands for card management
ISO/IEC 7816-10	1999	Electronic signals and answer to reset for synchronous cards
ISO/IEC 7816-11	2004	Personal verification through biometric methods
ISO/IEC 7816-12	2005	Cards with contacts USB electrical interface and operating procedures
ISO/IEC 7816-13	2007	Commands for application management in a multi-application environment
ISO/IEC 7816-15	2004	Cryptographic information application

The physical characteristics of the smart cards are regulated by the ISO 7816-1 and ISO 7816-2 standards. Figure 2.2 illustrates the dimensions of the smart card associated with these standards.

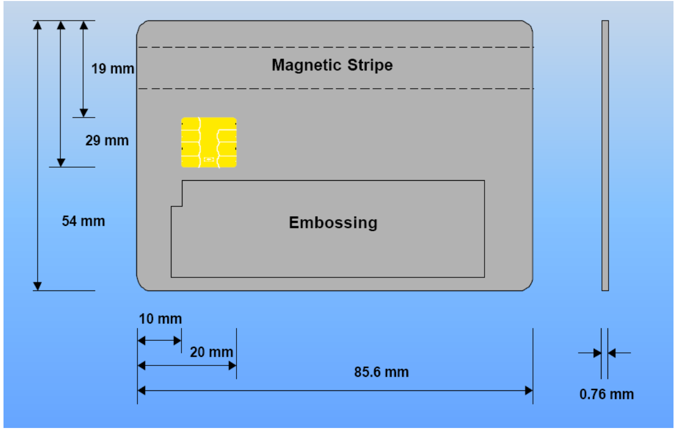


Figure 2.2: Smart card dimensions according to ISO/IEC 7816-1 and ISO/IEC 7816-2 (Jun, 2003).

The internal characteristics of a smart card include at least a Central Processing Unit (CPU), an input/output interface, a data bus and memory. Some smart cards contain a Public Key Infrastructure (PKI) coprocessor for RSA and ECC public key standards (NXP, 2008). In 1993 Philips Semiconductors reported the use of the P83C852 controller to generate or verify Digital Signatures using the public key algorithm RSA and also an example for the implementation of cipher functions related to DES (Imjela, 1996); currently, some card models include AES and Triple DES coprocessors (NXP, 2008).

Even today, smart card microcontrollers are most commonly based on the 8-bit CPU. The 8051 microcontroller is one of the most popular CPUs and has been on the market for more than two decades (Pearson and Albus, 2009). Until 2005, this microcontroller could be commonly found in car engine control units (Parab et al. 2007). The JCOP30 and JCOP41 Java Cards are two common smart cards which are used in this research. According to the database *smartcard_list* (Rousseau, 2012) these cards correspond to the Mifare ProX (Philips, 2003) and NXP JCOP41 (Philips, 2006) respectively. The Mifare ProX and Smart MX Java Cards are based on the extended 8051 architecture

(NXP, 2009c). There are some cards that use 16-bit microcontrollers and a few are based on 32-bit processor families such as ARM 7 or MIPS (Rankl, 2007).

2.1.2 Software

In the last few years, software within the smart cards has evolved from dedicated special purpose programs for a single application to true operating systems. One factor that has put pressure in this direction is the cost of custom tailored programs embedded in the ROM of the smart cards. The Java Card overcomes these obstacles allowing third party programmers to develop applications that can be installed and run in the card after the card leaves the production line. Java Cards offer an open platform that defines the standard application, programming interfaces and run time environment. The platform encapsulates the underlying complexity and details of the smart card system (ISO/IEC, 2011).

The smart card operating system supports a collection of instructions on which user applications can be built. For example, the ISO 7816-4 standard specifies a wide range of instructions in the form of application protocol data units (APDUs). Smart card operating systems may support some or all of these APDUs as well as the manufacturer's additions. Most smart cards support a modest file system based on ISO/IEC 7816-4 (Chen, 2000). This standard defines details such as the commands and responses transmitted by the card reader and vice versa; the content of the bytes sent during the answer to reset; the structure of files and data and the access methods to files and data in the card (ISO/IEC, 2011).

2.2 Types of Smart Cards

As mentioned before, there are three types of microprocessor smart cards: contact, contactless and dual interface. Each type of card has its own advantages. In the case of the contact cards, they have the advantage of computing with a faster processor and uninterrupted power supply. In the case of the contactless cards, they use radio frequency to power the card and also to communicate. These features are desirable in applications such as public transport where the absence of contact with the card reader prevent stress on both the card and reader and avoid failures due to bad contacts. In the case of the dual interface cards, they are able to work as contact and contactless cards with the contact interface mode preferred in the higher value transactions (Hendry, 2007).

2.2.1 Contact Cards

Contact smart cards come with contact eight pins. Six pins are used as follows: Power supply voltage (VCC), ground (GND), programming Voltage (VPP), input/output (I/O), reset (RST) and clock (CLK). The other two pins are reserved for future use. In the case of VCC, the typical voltage is 5 V. Figure 2.3 shows the specific position of each one of the pins.

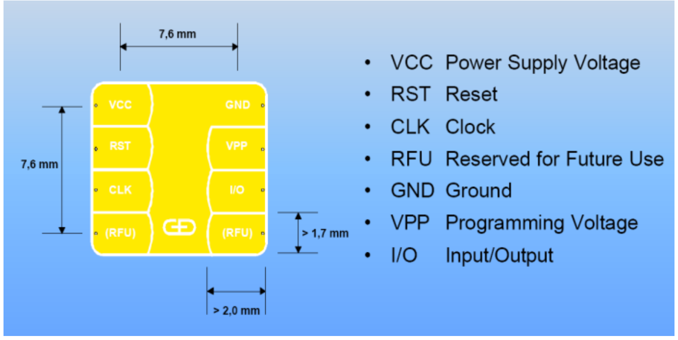


Figure 2.3: Smart Contacts of the Smart Card Module according ISO/IEC 7816-2 (Jun, 2003).

In the case of the contact cards, they typically contain a microprocessor (CPU), a coprocessor also called numeric process unit (NPU), ROM, EEPROM, and RAM. Figure 2.4 illustrates the internal organization of a contact smart card.

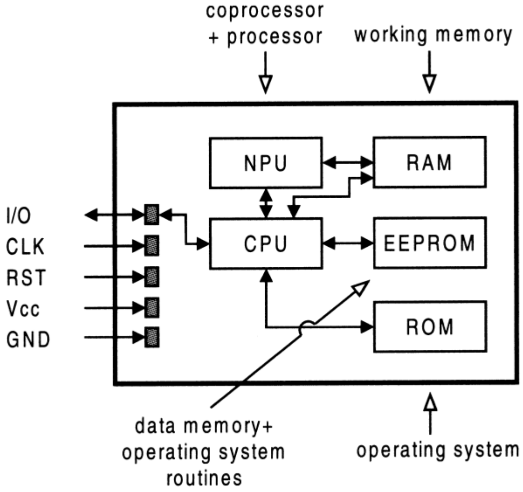


Figure 2.4: Typical architecture of contact microprocessor card (Rankl and Effing, 2000).

The amount of ROM, EEPROM and RAM depends on the microcircuit design; however, as a rule of thumb, each cell of RAM requires 4 times more space than a cell of EEPROM and this needs 4 times more space than a cell of ROM (Rankl and Effing, 2000). In the case of the ROM, it is used to store most of the operating system of the card. The data and programs in this type of memory are permanent and they can only be written during the manufacturing process. The EEPROM is used to store programs and information that should remain in the card after the power is off. The Java Cards store in this type of memory their Java applets. RAM is another type of memory, which stores temporal information and calculations of the CPU. The content of the RAM is lost once the card is powered off (Rankl and Effing, 2000).

2.2.2 Contactless Cards

Contactless cards do not have contacts and depend on the reader to power them. They work under the principle of inductive coupling and contain an antenna that is embedded in the card. This antenna is used to receive energy and also data from the terminal (reader). In accordance with the ISO 14443 standard, the reader or proximity coupling device (PCD) produces an energizing RF field of 13.56 MHz ± 7 kHz. The signal that comes from the antenna passes to the RF interface where the information is demodulated before it passes to the card's CPU. In the case of the response from the card to the reader, the information coming from the card's CPU is modulated and then sent to the reader through the antenna. Figure 2.5 shows the typical structure of this type of card. It is possible to observe that the card contains basically the same modules as the contact card but instead of the socket, it contains the RF interface.

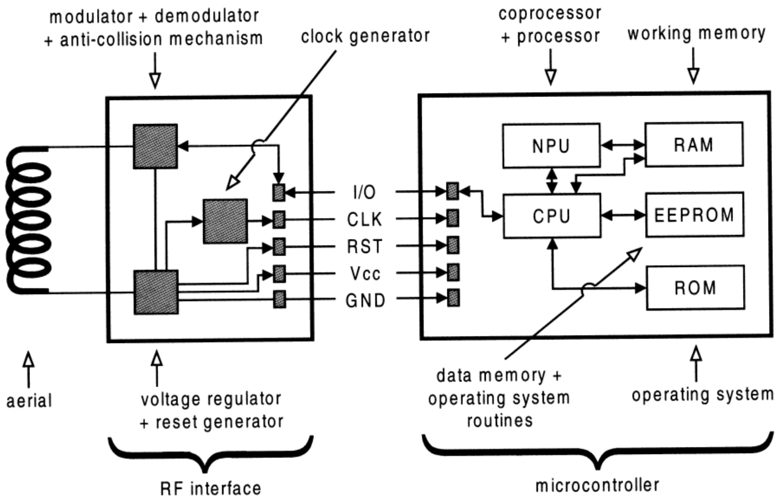


Figure 2.5: Typical architecture of a contactless card (Rankl and Effing, 2000). It includes a microcontroller and a RF interface in the same chip.

2.2.3 Dual Interface

Dual interface cards are both contact and contactless. Hence, their architecture is basically the same as a contactless card. On one side of the card they have the same contacts as a regular contact card and additionally embedded in the card but on the other side of the socket a connection to its antenna. Figure 2.6 presents a photo from a dual interface where the back of the card was scratched. The image shows the position of the chip and its antenna connections. Also visible are the contact points that come from the front of the card and how they are connected to the circuit. The position of the chip is essential for power analysis.

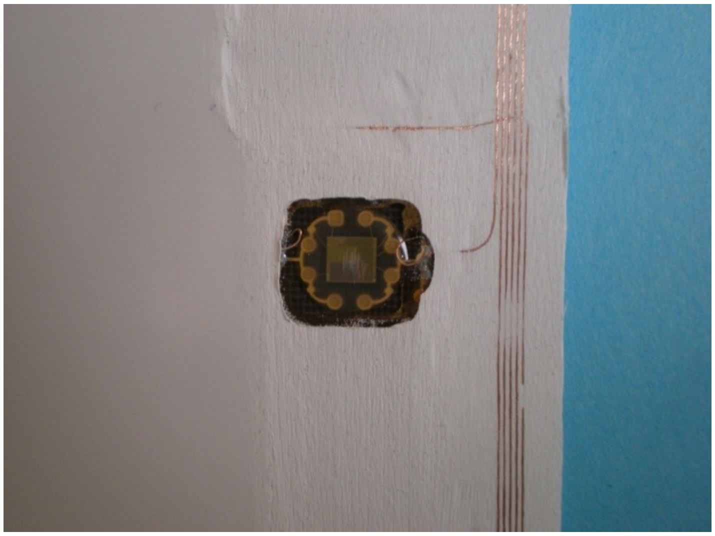


Figure 2.6: Microprocessor and antenna in a JCOP30.

As mentioned previously, the chip of the smart card contains the CPU, memory, RF module and all the other circuits necessary to work except the antenna that is embedded around the card. Memory resources are very important, not only in dual interface cards, but also in contact and in contactless cards. They restrict the size and number of applets that can be installed in the card. Similar to the contact and contactless cards, the dual interface cards come with ROM, EEPROM and RAM. Table 2.2 presents a comparative table showing the resources of 4 different types of dual-interface Java Cards, JCOP30, JCOP31-36, JCOP31-72 and JCOP41 (IBM, 2000) (IBM, 2002).

Table 2.2: Memory resources in different smart card models.

	JCOP30	**JCOP31-36**	**JCOP31-72** (Philips, 2004)	**JCOP41**
EEPROM:	14 kbytes	36 kbytes	72 kbytes	72 kbytes
RAM:	487 bytes	2,304 bytes	4,608 bytes	4,608 bytes
ROM:	24 kbytes	96 kbytes	160 kbytes	160 kbytes

2.3 Communication Protocols on Contactless Smart Cards

The communication protocols on contactless smart cards define how data is transmitted from/to a card reader to/from the card. The knowledge of these protocols allows one to determine the time when the card finishes receiving the data and when it starts transmitting the computed response. This processing time in the microprocessor is fundamental to setup a side channel attack.

The ISO 14443-2 standard (ISO/IEC, 2011) establishes the communication and transmission protocols between card readers and contactless smart cards. It may follow two types of communication, type A or type B. All card readers must support both types of protocols, even though the cards only need to use one of them at a time.

2.3.1 Type A Protocol

The protocol known as type A, uses amplitude shift keying (ASK) modulation 100% with modified Miller coding for communication from the card to the reader. As its name indicates the amplitude of the carrier oscillation is switched between 100% and 0%. In the modified Miller coding, a “1” is represented by a “pause” (in the half bit period). For a logic “0”, no modulation shall occur for the full bit duration except if there are two or more contiguous “0”s. In that case, a “pause” shall occur from the second “0” at the beginning of the bit period (Finkenzeller, 2003). The blanking intervals or pauses go from 2 μs to 3.5 μs. This coding guarantees a continuous power supply from the reader to the contactless card. Figure 2.7 shows how the ones and zeros are modulated using this technique.

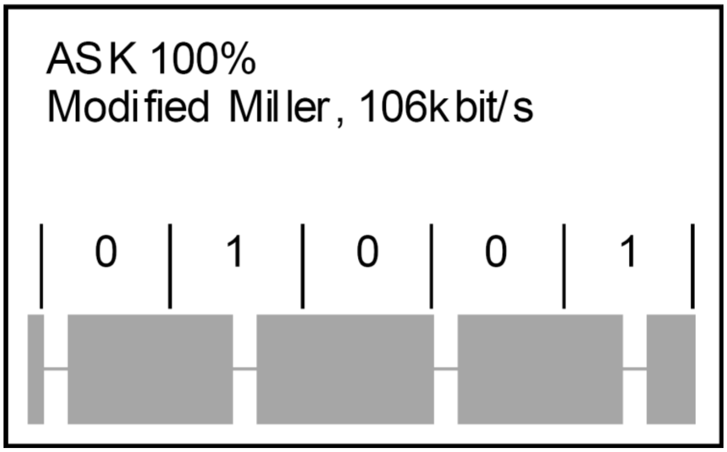


Figure 2.7: Modulation ASK 100% with Modified Miller coding (ISO/IEC, 1999a).

In the type A protocol, the response from the card to the reader uses load modulation with a subcarrier. The subcarrier frequency is 13.56 MHz/16 ≈ 847 kHz. For the subcarrier modulation every bit period starts with a defined phase relation to the subcarrier. The bit period starts with the loaded state of the subcarrier. Figure 2.8 shows the way this modulation is performed. For a logical “1”, the carrier is modulated with the subcarrier for the first half of the period. For a logical “0”, the carrier is modulated with the subcarrier for the second half of the period. The type A protocol is used in the Java Cards analyzed in this research.

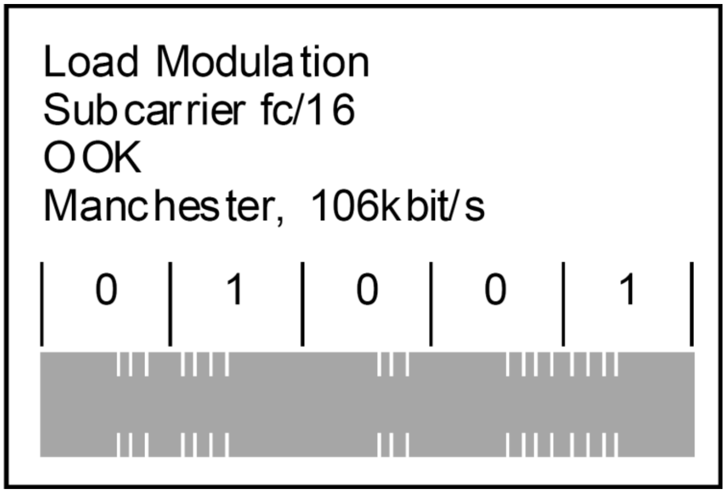


Figure 2.8: Load modulation with subcarrier using Manchester coding (ISO/IEC, 1999a).

2.3.2 Type B Protocol

The type B protocol is specified in the ISO 14443-2 standard. It uses ASK 10% modulation for communication from the card to the reader. As its name suggests the modulation index should be 10% with a minimum of 8% and a maximum of 14% accepted (ISO/IEC, 1999a). The coding used in this protocol is Non-Return-to-Zero-Level (NRZ-L). For logic “1”, no modulation is applied and for “0” the carrier uses the low field amplitude. Figure 2.9 presents an example of this coding.

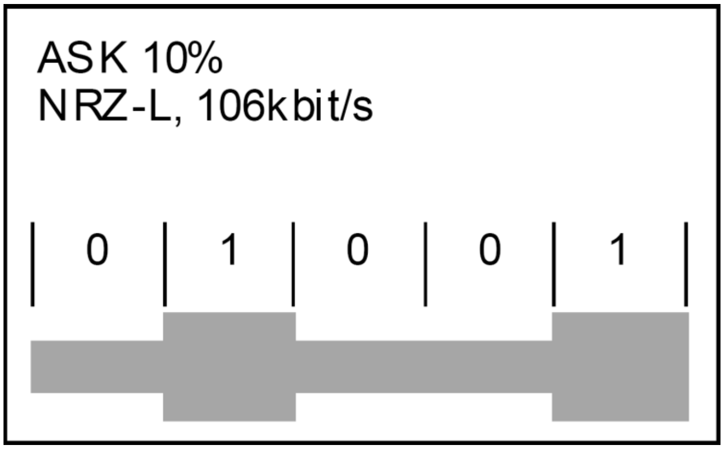


Figure 2.9: Modulation ASK 10% amplitude using NRZ-L coding (ISO/IEC, 1999a).

The response from the card to the reader for the type B protocol uses load modulation with a subcarrier of 847 kHz. The subcarrier is modulated by 180˚ phase shift keying (BPSK) of the subcarrier using the NRZ coded data stream (Finkenzeller, 2003). To start a transmission from the card to the reader, the card generates a subcarrier phase reference $ø_0$. This initial phase state $ø_0$ of the subcarrier is defined as logical 1 so the first phase transition represents a change from logical “1” to logical “0”. Next, in accordance with the phase reference, the logical “1” is $ø_0$ and the logical “0” is $ø_0$ + 180˚. Figure 2.10 shows an example of this response.

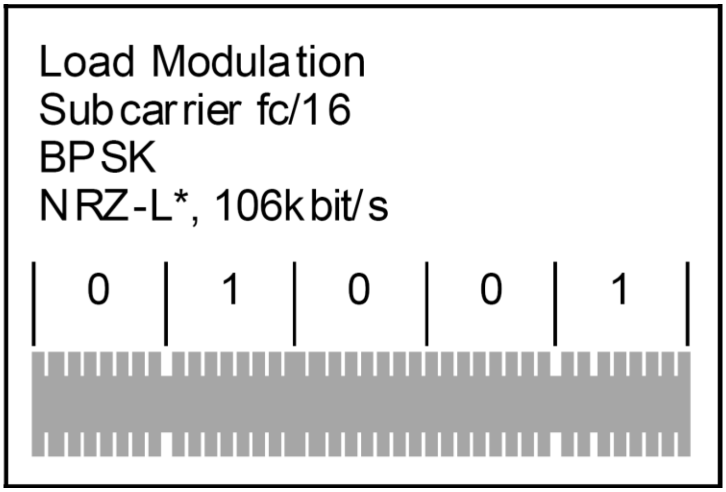


Figure 2.10: Load Modulation with subcarrier, using BPSK NRZL coding (ISO/IEC, 1999a).

2.4 Security in Smart Cards

As mentioned before to increase performance, some cards contain special coprocessors to encrypt or decrypt information using a symmetric key scheme. Software implementations are cheaper than hardware implementations but usually are slower. For example, in this research an applet was implemented that encrypts a 128-bit message using the Advanced Encryption Standard (AES). In a JCOP30 card, the AES algorithm took approximately 3 seconds to encrypt the information. Hardware implementations are faster than software but they are rigid and cannot be changed once they leave the production line. Java Cards have such flexibility that it is possible to change and update the programs at any time.

There are many encryption/decryption algorithms used in smart cards; some of them are public and well known like the Data Encryption Standard (DES), Triple-DES, or AES. Others are "secret" as in the case of CRYPTO1 (Koning et al. 2008). AES will be described next since it will be the focus of the attack in this research.

2.4.1 AES

AES is the acronym for the Advanced Encryption Standard. This standard was announced on November 26, 2001 by the National Bureau of Standards now known as the National Institute of Standards and Technology (NIST). It came into effect on May 26, 2002 and came to replace DES for unclassified information requiring cryptographic protection (NIST, 2001).

In some smart cards, this standard is implemented in hardware such as JCOP41 cards, but AES can also be implemented in software. AES is a symmetric block cipher that uses sequences of 128 bits and may use keys with lengths of 128 bits, 192 bits or 256 bits. Depending on the key length, it is sometimes called AES-128, AES-192 or AES-256.

At the start of AES, the input is copied to a "State" array. Next, depending on the key length, a routine named Key Expansion is used to calculate the array "w". It is followed by 9, 11 or 13 rounds of transformations depending on the key length plus a final round that is slightly different (MixColums transformation). In each round, the algorithm does a SubBytes (S-box) transformation, a ShiftRows transformation, a MixColums transformation, and an AddRoundKey. The pseudocode for the AES cipher is shown in Figure 2.11.

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
Begin
    byte state[4,Nb]
    state = in

    AddRoundKey(state, w[0, Nb-1])

    for round = 1 step 1 to Nr-1
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    end for

    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
    out = state
end
```

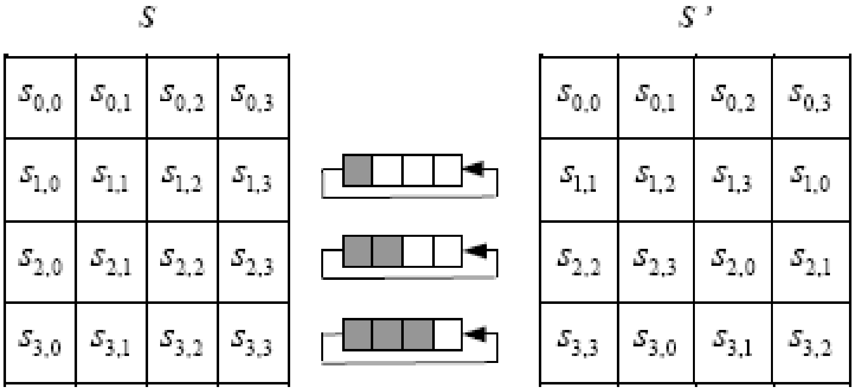
Figure 2.11: Pseudocode for the AES cipher (NIST, 2001).

The SubBytes transformation consists of a non-linear substitution, where each byte of the State is replaced by a corresponding value using the S-box table (Table 2.3).

Table 2.3: S-box: substitution values for the byte *xy* in hexadecimal format (NIST, 2001).

		Y															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
x	**0**	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

The ShiftRows transformation consists of cyclically shifting the last three rows of the State. Figure 2.12 illustrates how this transformation works.



S
S '
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,1}$ $s_{1,2}$ $s_{1,3}$ $s_{1,0}$
$s_{2,2}$ $s_{2,3}$ $s_{2,0}$ $s_{2,1}$
$s_{3,3}$ $s_{3,0}$ $s_{3,1}$ $s_{3,2}$

Figure 2.12: ShiftRows() cyclically shifts the last three rows in the State (NIST, 2001).

The MixColums transformation operates on the State array and treats each column as a four term polynomial over GF(2^8) which is multiplied modulo x^4+1 with a fixed polynomial *a(x)* given by (2-1)

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} \tag{2-1}$$

The AddRoundKey transformation consists of a bitwise XOR operation between the State array and a word from the key scheduled for that round. Figure 2.13 shows the pseudocode to calculate the key expansion.

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
   word temp
   i = 0
   while (i < Nk)
      w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
      i = i+1
   end while
   i = Nk
   while (i < Nb * (Nr+1)]
      temp = w[i-1]
      if (i mod Nk = 0)
         temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
      else if (Nk > 6 and i mod Nk = 4)
          temp = SubWord(temp)
      end if
      w[i] = w[i-Nk] xor temp
      i = i + 1
   end while
end
```

Figure 2.13: Pseudo code for Key expansion (NIST 2001).

SubWord() is a function that takes a four-bit word and applies the S-box table. RotWord() is a function that also takes a four-bit word and performs cyclic permutations to the values of the word and Rcon(i) is the round constant word array; it contains the values given by: [02^{i-1}, {00}, {00}, {00}].

Implementations of this algorithm are typically attacked on the S-box. The preferred targets are the first or last AES S-box. In the case of the first S-box in round one, the intermediate result is a function of the first byte of plaintext and the first byte of the secret key while in the last round it is function of the ciphertext and the last byte of the secret key (Mangard et al. 2007). The next chapter introduces different types of cryptographic attacks.

2.5 Summary

A brief overview of the hardware and software characteristics of microprocessor-based smart cards in this chapter has been presented. The communication protocols type A and type B were described. The type A protocol is used by the Java Cards studied in this research. This protocol is employed by the card reader to communicate with the smart card and is used in Chapter 6 for triggering the oscilloscope. The AES algorithm was also briefly outlined at the end of the chapter and it is used by the different systems tested in this research. The next chapter presents an overview of side channel analysis and focuses on correlation analysis.

Chapter 3

Side Channel Analysis

Side channel analysis is the process of obtaining additional information about the internal activity of a physical device beyond that allowed by its specifications. Timing attacks are an example of side channel attacks. This kind of attack was introduced by Paul Kocher to attack implementations of Diffie-Hellman, RSA, DSA and other cryptographic algorithms. It consisted of measuring the time required to perform private key operations to determine, for example, in the Diffie-Hellman scheme, the Diffie-Hellman exponents (Kocher, 1996). Power analysis and EM analysis are types of side channel attacks used to look for weaknesses in the algorithm implementation. The main idea is to measure the power consumption or EM emanations of a device when it is encrypting/decrypting a plaintext/ciphertext using a secret key, and then analyse those measurements to try to recover the secret key.

This chapter describes the characteristics of power analysis and EM analysis. It focuses on one specific type of attack called correlation analysis and explains how this attack is implemented in the time and in the frequency domain. It also presents two new ways to evaluate the results of a side channel attack. At the end of this chapter, some preliminary results show the effectiveness of the proposed techniques (Mateos and Gebotys, 2010). These results include the use of EM and power traces in the time and frequency domains. The EM traces were acquired with a commercial inductive probe placed over the 8-bit AT89C51ED2 microcontroller which is based on the Intel family 8051. This microcontroller was selected considering its relative similarity to the extended 8051 architecture (NXP, 2009c) inside the JCOP30 and JCOP41 Java Cards, which are later studied in this research. The effects of the clock frequency and number of traces used in the correlation analysis in the frequency domain are explored. The analyses in the time and frequency domain are compared using the proposed metrics, the execution time, and the capacity to deal with random misalignments. The power traces used to complement the study come from the public database of the DPA book WS3 (Mangard et al. 2010b).

This workspace includes two sets of 1,000, traces one with dummy operations and one without them, and correlation analysis in the time and frequency domain is used to analyse them.

3.1 Power Analysis

Simple power analysis (SPA) is an example of side channel analysis where an attacker monitors the power consumption of the processor when it executes a cryptographic operation and directly interprets the power measurements (Kocher et al. 1999). Using one or more power traces, the attacker may identify characteristics such as timing, device attributes and the implemented algorithm structure (Kocher et al. 2011). With simple power analysis, the attacker can look for branching operations or distinguish between operations with different power consumptions (Kocher et al. 2004) (Gebotys, 2010). Simple power analysis has been used to attack cryptographic implementations in smart cards. For example, research in (Novak, 2002) presents an attack on a smart card implementation of the RSA decryption algorithm. There are some cases where the power consumption of a simple instruction has a direct correlation with the Hamming weight of the data processed. This fact was shown in (Mayer-Sommer, 2000) using a PIC16C84 microprocessor and also in (Messerges et al. 2002) using an HC05 microprocessor based smart card.

Differential Power Analysis (DPA) is a more powerful kind of side channel analysis which was introduced by Kocher, Jaffe and Jun (Kocher et al. 1999). This technique looks for weaknesses in the algorithm implementations and presents some advantages over SPA, because DPA does not require detailed knowledge of the device under study. This method is based on statistical analysis and generally requires a large number of samples and traces. The attack focuses on guessing an intermediate result inside the algorithm that is a function of the plaintext/ciphertext and the secret key. First, the power consumption from the device under scrutiny is recorded when the device is encrypting/decrypting data. The device encrypts/decrypts data *m* times and the attacker records *m* traces. Let the power trace obtained at the *d-th* iteration be T_d and $T_d(j)$ is the *j-th* sample within this trace. Then, the traces are divided into two groups according to the value of the *g-th* bit (b_g) of intermediate data calculated from a key guess. One group called $T_{one}(t)$ contains the d_{one} traces where the bit b_g is expected to be 1, $T_{one}(t)=\{T_d(t)| \forall d$ where b_g="1"}. The other called $T_{zero}(t)$ contains d_{zero} traces where the bit b_g is expected to be "0", $T_{zero}(t)=\{T_d(t)| \forall d$ where b_g="0"}. Next the mean of each group of traces is calculated $\left(\overline{T_{one}(t)}, \overline{T_{zero}(t)}\right)$ and subtracted from each other. The difference of means (*DOM*) attempts to eliminate the algorithm impacts on the power traces and keeps the information related to bit *g* (Gebotys, 2006). The differential analysis based on one guess of the key is calculated using (3-1)

$$DOM(t) = \overline{T_{one}(t)} - \overline{T_{zero}(t)} \tag{3-1}$$

The maximum magnitude over all key guesses is used to determine the correct key. DPA was used to examine the security of smart cards against DES implementations as studied in (Kocher et al. 1999), (Messerges, 2000), (Clavier et al. 2000), and (Messerges et al. 2002). In a similar way, DPA was applied to attack an elliptic curve (EC) cryptosystem as shown in (Coron, 1999) where they also showed the vulnerabilities of unprotected implementations of EC Diffie-Hellman key exchange and EC El-Gamal encryption. In (Messerges et al. 1999) DPA was used against smartcard implementations of modular exponentiation while in (Lu et al. 2009b) it is used to analyse an ASIC implementation of AES. A variation of DPA call Differential ElectroMagnetic Analysis (DEMA) was suggested by (Quisquater and Samyde, 2001) to analyse a contact smart card. With DEMA instead of using power traces one uses electromagnetic traces. DEMA was used to analyse the security of other smart cards in (Gandolfi et al. 2001) and (Agrawal et al. 2002).

In (Chari et al. 2002), template attacks were introduced. This attack involves two phases; first, the attacker characterizes the device by building a template model, executing a determined sequence of instructions using fixed data and capturing the corresponding power or EM traces. With these traces, the attacker tries to build a precise multivariate normal distribution model (Mangard et al. 2007). The second phase consists of matching the template models with the traces acquired from the device under attack (Medwed and Oswald, 2008).

Mutual Information Analysis (MIA) is a generic side channel attack based on information theory distinguishers. It was proposed by (Gierlichs et al. 2008) and measures the total dependency between two random variables X and Y. The total information shared between the two variables can be expressed in terms of entropy using Shannon's formula I(X;Y) = H(X)–H(X|Y), where H(X) is the entropy of the random variable X and H(X|Y) is the conditional entropy of the random variable X given the variable Y (Menezes et al. 1997) (Veyrat-Charvillon and Standaert, 2009). Although, MIA was expected to exploit all information contained within trace measurements and capture the true dependency between the real device leakage and the modeled leakage, there is little evidence that these expectations are met in practice (Whitnall and Oswald, 2011). Different works like (Prouff and Rivain, 2010) have compared Correlation Power Analysis to MIA and the results indicate that MIA is less efficient than CPA when the deterministic part of the leakage is a linear function of the model used by the attacker. In (Moradi et al. 2009) it was found that MIA had a bigger computational overhead than CPA and MIA worked worse in the presence of noise. Research in (Veyrat-Charvillon and Standaert, 2009) indicated that when a reasonable leakage model is known, techniques such as Correlation Power Analysis are more efficient than MIA.

There are other types of side channel analysis and a good review can be found in the DPA book (Mangard et al. 2007). Research in (Doget et al. 2011) compared different side channel attacks. The attacks analysed included Differential Power Analysis (Kocher et al. 1999), Enhanced DPA (Bevan and Knudsen, 2003), Correlation Power Analysis (Brier et al. 2004), and Partitioning Power Analysis (PPA) (Thanh-Ha Le et al. 2006). The result of this study concluded that if the attacker has a good linear approximation of the leakage function, Correlation Power Analysis (CPA) is an optimal way to perform an attack.

3.2 EM Analysis

Electromagnetic (EM) analysis is similar to power analysis; however, the EM emanations from an electronic device are measured instead of power signals. The EM emanations are recorded when the device is encrypting or decrypting a plaintext/ciphertext using a secret key. Similar to DPA, the attacker later tries to recover the key through analysing the recorded measurements. EM analysis is based on some principles of EM theory that are briefly described below.

In the first place, Biot-Savart's law states that when an electrical current i_c moves through a straight conductor, at all points on a circle of radius r around that conductor, the magnitude of the magnetic field $\vec{B}$ generated is given by Equation (3-2), where μ_0 is the permeability of free space (μ_0=4π×10^{-7} Wb/A·m) (Young and Freedman, 1996).

$$B = \frac{\mu_0 i_c}{2\pi r} \tag{3-2}$$

In a more generic way, Maxwell's third equation (3-3) refers to a similar phenomenon. It states that the magnetic field $\vec{B}$ depends on the conduction current i_c and the displacement current $\epsilon_0 d\Phi_E/dt,$ where ϵ_0 is the permittivity of free space ($\epsilon_0 \approx$ 8.85 x 10^{-12} F/m) and $d\Phi_E/dt,$ is the time rate of change of electric flux (Young and Freedman, 1996).

$$\oint \vec{B} \cdot \overrightarrow{dl} = \mu_0 \left(i_c + \epsilon_0 \frac{d\Phi_E}{dt} \right) \tag{3-3}$$

In this scenario, one might assume that when an electronic device is processing information, some currents are flowing inside the internal circuits and consequently producing magnetic fields.

Another principle exploited by EM analysis is Faraday's law (3-4), which states that the induced electromotive force ε in a closed loop equals the negative of the time rate of change of magnetic flux $d\Phi_B$ through the loop. The term $d\Phi_B$ equals the magnetic field $\vec{B}$ for an infinitesimal area $\overrightarrow{dA}$ (Young and Freedman, 1996).

$$\varepsilon = -\frac{d\Phi_B}{dt} \tag{3-4}$$

In a similar way, Maxwell's fourth equation (3-5) states that a changing magnetic flux $d\Phi_B$ induces an electric field $\overrightarrow{E}$ (Young and Freedman, 1996).

$$\oint \vec{E} \cdot \overrightarrow{dl} = -\frac{d\Phi_B}{dt} \tag{3-5}$$

For this reason, every time a coil is placed close to a magnetic field, it induces an electromotive force on the coil. Specifically placing a coil or an EM probe close to an electronic circuit that is processing data, might induce some voltages on the terminals of the probe that are correlated with the data processed by the circuit and consequently provide information about the data itself.

3.3 Correlation Analysis

The population correlation coefficient ρ is defined as a measure of linear association or clustering around a line (Freedman et al. 1998). Considering two sets of data X and Y with population standard deviations σX, σY and population covariance $cov(X,Y)$, the magnitude of the population correlation coefficient ρ could be calculated using (3-6). When the ρ is close to 1 or -1, there exists a strong correlation between the sets, when ρ is close to zero there is a weak correlation.

$$\rho(X,Y) = \frac{cov(X,Y)}{(\sigma X)(\sigma Y)} \tag{3-6}$$

In the case of side channel analysis, the correlation coefficient ρ indicates the similarity between the hypothetical power consumption model and the one measured (power consumption or EM power consumption). In general, the value of the population correlation ρ is unknown and needs to be estimated by using the sample covariance and sample standard deviation (Mangard et al. 2007). This estimate of the correlation factor ρ can be denoted by $\hat{\rho}$ (Brier et al. 2004) or similar to this research, r is used. In the next sections, three variations of correlation analysis, in the time domain, in the frequency domain, and one called fast correlation in the frequency domain are described. All descriptions are matrix based which allow a straightforward implementation.

3.3.1 Correlation Analysis in the Time Domain

Correlation of coefficients is a method used to measure the correlation between the power traces obtained when an electronic device is processing a given data and the hypothetical power consumption model. The hypothetical power consumption model can be the Hamming-weight HW or the Hamming distance of the data processed. It was first proposed by (Brier et al. 2004) and since then, it has been widely studied on a variety of devices such as 8-bit microcontrollers running AES implementations (Mangard, 2004), (Doget et al. 2011), application-specific integrated circuits (ASICs) (Ors et al. 2004), (Lu et al. 2009a), microcontrollers with AES coprocessors (Kizhvatov, 2009), FPGAs using DES implementations (Standaert et al. 2004) or CAST-128 implementations (Boey et al. 2010), (Kean et al. 2010). Also EM correlation analysis in the time domain has been studied with DES implementations in smart cards (Kasper et al. 2011), (Kasper et al. 2009), (Oswald and Paar 2011), and ASICs (Thanh-Ha Le et al. 2006), (Le et al. 2008).

Correlation analysis works as follows: first, it is necessary to define an attack point in the algorithm implementation. This point must be a function of non-constant data and a small part of the key. Using *D* different plaintexts/ciphertext and considering all possible values of a small portion of the key *K*, the hypothetical power consumption model matrix *HP* is prepared. The Hamming distance can be used to build the power consumption model. The Hamming distance corresponds to the number of bits that differ from an initial value v_0 and its final value v_1 HD(v_0,v_1). The Hamming weight is used some times to model the power consumption. It corresponds to a particular case of Hamming distance where the initial value v_0=0. Hamming weight models are typically more suitable for circuits that contain pre-charged buses.

The Pearson correlation of coefficients requires the measurement or acquisition of *D* power consumption or EM traces. Let $T_d(j)$, represent the *j-th* sample in the *d-th* trace, where j=1,...N for *N* samples per trace and d=1,...D. Using Equation (3-7), the Pearson correlation for the matrix of correlation *R* is calculated (Brier et al. 2004). Additional terminology includes: K_i, i=1,…,k, for the *i-th* hypothetical value of the small part of the key *K;* $\overline{T(j)}$ corresponds to the mean of all *D* traces, $T_d(j)$, at the *j-th* sample; HP_i is the hypothetical power or Hamming weight of the intermediate data for the *i-th* hypothetical key guess; $\overline{HP_\iota}$ corresponds to the average hypothetical power for the hypothetical *i-th* key guess (or mean of all *D* Hamming weights for the *i-th* key guess).

$$r_{i,j} = \frac{D\sum_{d=1}^{D}\left(T_d(j)\cdot HP_{d,i}\right) - \sum_{d=1}^{D}\left(T_d(j)\right)\cdot\sum_{d=1}^{D}\left(HP_{d,i}\right)}{\sqrt{D\sum_{d=1}^{D}\left(T_d(j)-\overline{T_d(j)}\right)^2 \cdot D\sum_{d=1}^{D}\left(HP_{d,i}-\overline{HP_\imath}\right)^2}} \tag{3-7}$$

The possible correct small part of the key, K_i , used for the encryption/decryption of the data corresponds to row *i* that contains the element with the maximum magnitude (positive or negative).

3.3.2 Correlation in the Frequency Domain

Correlation analysis in the frequency domain measures the level of association between the hypothetical power consumption model and the power spectrum measured. It requires the power spectrum F_d from the *D* power or EM traces using the Fast Fourier Transform (FFT) as indicated in Equation (3-8). Here $F_d(j)$ represents the amplitude of the frequency *j* for trace *d,* where *j*=1,...,*NFFT*, *NFFT* is the number of frequency components returned by the FFT transform and is related to the sampling rate, and *d*=1,..*D,* is the *d-th* trace used in the analysis. Since the FFT is a symmetrical transformation, it is possible to analyze only one-half of the frequencies. Thus, the size of $F_d(j')$ becomes *D* rows and *NFFT*/2 columns, with *j'=1,...,NFFT/2.*

$$F_d(j') = |FFT(T_d)|^2 \tag{3-8}$$

The hypothetical power consumption model is generated following the same steps as in the analysis for the time domain. The Pearson correlation *r* between the hypothetical power consumption model and the power spectrum is calculated using Equation (3-9). Equations (3-7) and (3-9) have the same structure, with the differences that (3-7) uses the power or EM traces $T_d(j)$ in the time domain and it's of size [*D*×*N*]; while Equation (3-9) utilize the power spectrum $F_d(j')$ and it is of size [*D*×*NFFT*/2].

$$r_{i,j'} = \frac{D\sum_{d=1}^{D}\left(F_d(j')\cdot HP_{d,i}\right) - \sum_{d=1}^{D}\left(F_d(j')\right)\cdot\sum_{d=1}^{D}\left(HP_{d,i}\right)}{\sqrt{D\sum_{d=1}^{D}\left(F_d(j')-\overline{F_d(j')}\right)^2 \cdot D\sum_{d=1}^{D}\left(HP_{d,i}-\overline{HP_\imath}\right)^2}} \tag{3-9}$$

Each row in the correlation matrix *R* is associated with a possible key guess (*i*) and the columns are related to the frequencies from the power spectrum used *(j').* As in the time domain analysis, the possible correct key is determined by finding the row *i* that contains the element with the maximum absolute values from the matrix.

Correlation analysis in the frequency domain has been used to analyse AES simulation power traces from a 0.25 µm CMOS design (Schimmel et al. 2010), EM traces from 8-bit microcontrollers (Peng et al. 2009), and EM traces from FPGAs (Hodgers et al. 2011). In a similar way, has been used to analyse EM traces from hardware implementations of DES in smart cards using customised analog demodulation (based on low-pass active filters) that removed the carrier and high frequency components (Kasper et al. 2011), (Oswald and Paar, 2011).

3.3.3 Proposed Fast Correlation in the Frequency Domain

Fast correlation in the frequency domain is based on correlation analysis in the frequency domain, but the main difference is the smaller number of frequencies, *P,* considered in the analysis. It requires pre-characterizing the attacked device to determine the range of frequencies that are more likely to return higher correlations or the range of frequencies more likely to relate to the data under attack. An empirical analysis is required to determine the range or set of frequencies with the higher magnitudes, when most of the power or EM measurements from the device are associated with the encryption/decryption process.

This range of frequencies (or equivalently set of columns) is represented by j'', where $j''=m+1, m+2, \ldots , m+P$ and $\{m+1, m+2, \ldots , m+P\}\subset\{1, 2, \ldots , \text{NFFT}/2\}$. Let $F_d(j'')$ represent the amplitude of the j''*-th* frequency in the *d-th* trace. The new matrix *F* and the hypothetical power consumption model are used to calculate the matrix of correlation using the Pearson correlation defined by Equation (3-10).

$$r_{i,j''} = \frac{D\sum_{d=1}^{D}\left(F_d(j")\cdot HP_{d,i}\right)-\sum_{d=1}^{D}\left(F_d(j")\right)\cdot\sum_{d=1}^{D}\left(HP_{d,i}\right)}{\sqrt{D\sum_{d=1}^{D}\left(F_d(j")-\overline{F_d(j")}\right)^2\cdot D\sum_{d=1}^{D}\left(HP_{d,i}-\overline{HP_{\imath}}\right)^2}} \qquad (3\text{-}10)$$

The matrix of correlation *R* contains *P* columns and *i* rows. Each column is associated with one of the frequencies selected and each row is related to one of the possible key guesses. The value from the matrix with maximum magnitude corresponds to the possible correct key. If the guessed key is equal to the correct key, the range of frequencies is kept as a possible set. When the guessed key is different from the correct key, then the range of frequencies is modified to include other frequencies and the analysis repeated. In case the correct key is unknown, Section 3.4 presents a technique to evaluate the significance of the key guessed.

The number of operations required to perform the FFT depends on the size of *N*. When *N* is a power of 2, the data transformation requires $N \log_2 N$ operations, otherwise it would require N^2 operations (Storey, 2002). The number of operations required for computing the different correlation analysis would vary depending on the size of *N* and the range of frequencies used in the analysis. In general, correlation analysis in the time domain requires one to correlate a matrix *HP* of size [$D\times i$] with another matrix *T* of size [$D\times N$]. It is necessary to calculate the correlation of $i\times N$ vectors of length *D.* In the case of correlation analysis in the frequency domain, the size of the matrix *F* is set to [$D\times NFFT/2$], where *NFFT* is recommended to be equal to the next power of 2 that is greater than or equal to *N*, and *NFFT/2* is always smaller or equal to *N.* Hence, correlation analysis in the frequency domain processes fewer than or an equal number of correlation vectors to the time domain analysis when all the samples are processed. Fast correlation in the frequency domain correlates fewer vectors than correlation analysis in the frequency domain analysis because only a subset of *NFFT'* frequencies is analyzed. One aspect to consider to improve the performance of the fast correlation in the frequency domain is to analyse a small number frequencies, *NFFT',* thus the overhead time required to obtain the *D* FFT is smaller than the time required to calculate the correlation of $i\times(N\text{-}NFFT')$ vectors of size *D.*

3.4 Evaluating the Effectiveness of an Attack

Few metrics have been proposed to measure the effectiveness of a side channel attack. Two security metrics that evaluate the success rate and guessing entropy are described in (Standaert et al. 2009). The success rate metric works as follows: First, all key guesses are sorted from most likely to least likely. To find the success rate of order *0,* one needs to determine if the correct key is sorted among the first *0-th* ranks. If this is true, the success rate equals 1 otherwise the success rate equals 0. A success rate of order 1 means that the correct key guess was sorted in the first rank. The guessing entropy metric presented in (Kopf and Basin, 2007) and (Standaert et al. 2009) is based on the guessing and entropy work published by (Massey, 1994). In a simplified way, the guessing entropy works as follows: When all key guesses are sorted from the most likely to least likely, the guessing entropy corresponds to the rank position where a guessing key corresponds to the correct key. This value represents the expected number of key candidates that need to be tested after the side channel attack (Standaert et al. 2009).

Sometimes the effectiveness of the attack is measured considering the number of traces required for discovering the correct key. Equation (3-11) proposed in (Mangard, 2004) illustrates a way to calculate the number of traces S necessary to distinguish a significant peak in the correlation matrix in the time domain, where the quantile z_α determines the distance between distributions with the maximum correlation $\rho_{\max}$ and noncorrelated ρ=0.

$$S = 3 + 8\left(\frac{z_\alpha}{\ln\left(\frac{1+\rho_{max}}{1-\rho_{max}}\right)}\right)^2 \qquad (3\text{-}11)$$

Another important parameter to consider when evaluating side channel attacks is the level of the noise. In (Mangard et al. 2007, Mangard, 2004), the authors use Equation (3-12) to measure the noise floor in correlation analysis based on the number of traces D.

$$noise\ floor = \frac{4}{\sqrt{D}} \qquad (3\text{-}12)$$

In this thesis, two new metrics are proposed that evaluate the effectiveness of an attack. One is called *accuracy* and the other *estimation*. Similar to the guessing entropy described in (Standaert et al. 2009), accuracy quantifies how close the key guess is from the correct key. It requires knowledge of the correct small part of the key but unlike the guessing entropy, it uses an exponential scale that minimises the contribution of the key guesses ranked in the last place and increases the weight on the guesses ranked in the first position. Equation (3-13) illustrates this calculation of the accuracy. The accuracy will change depending on the position the key guess is ranked. For example, first place corresponds to an accuracy of 1, second place to an accuracy of 0.5, third place an accuracy of 0.25, etc. The main rationale is to evaluate an attack that reaches the correct key as two times better than one that returns the correct key in second place, and four times better than another one that returns the correct key in third place.

$$accuracy = \frac{1}{2^{rank-1}} \tag{3-13}$$

The estimation metric is similar to the success rate of order 1 (Standaert et al. 2009) since it returns a 1 when the correct key is found and 0 if the correct key is not found. However, unlike the success rate, in the case of estimation the attacker does not need to know the correct key in advance. The estimation metric is based on statistical analyses of the correlation matrix and it provides a measure of confidence in the current key guess. Thus, it is useful for actual attacks as well as research analysis. Estimation is based on the magnitude of the correlation coefficients and how dispersed those values are. It measures the distance between the absolute maximum values from the correlation matrix *R* and the arithmetic mean of the maximum values for each key guess. The result is normalized dividing the distance between the maximum correlation value and the mean from all maximums by the standard deviations from all maximums. The way to obtain the estimation is shown in Algorithm 3.1 (Mateos and Gebotys, 2010).

Algorithm 3.1: Estimation

Input: Correlation matrix R, coefficient δ

Output: Estimation value {0 or 1}

```
$for\ each\ i = 1, \dots k, (j = 1, \dots, N)$
$Max\_R_i \leftarrow Maximum(abs(R_{i,j}))$
$Mean\_R \leftarrow mean(Max\_R_i)$
$STD\_R \leftarrow STD(Max\_R_i)$
$r \leftarrow Maximum(Max\_R_i)$
$if (r - \delta * STD\_R) > Mean\_R$
        $estimation \leftarrow 1$ ,
$else$
        $estimation \leftarrow 0$
$Return\ estimation$
```

The result from the *estimation* is a binary value that helps to decide if the analysis was meaningful or not. Accuracy and Estimation are additive metrics so that the results obtained for each small part of the key may be added to determine the effectiveness of the attack for the whole key. In the case of an attack on AES 128 where the small part of the key is 1 byte, an accuracy=16 means the whole key was successfully founded and an *estimation*=16 indicates that the returned key is meaningful for a 16 byte key algorithm.

Unlike previous metrics, the use of an exponential scale in the accuracy metric helps the security researcher to more easily discriminate and evaluate good attacks, where the whole correct key was found or the marginal guesswork (Kopf and Basin, 2007) is little, from bad attacks, where marginal guesswork is big. In the case of the estimation metric, one can evaluate the effectiveness of an attack without knowing the correct key, based only on the statistical properties of the resulting correlation matrix.

The next section compares the three correlation analysis methods described in Section 3.3 using different sets of traces such as EM traces using a commercial inductive probe and the power traces published in the DPAbook website (Mangard et al. 2010a). In some of the comparisons presented, the metrics of accuracy and estimation are used.

3.5 Experimental Setup and Initial Preliminary Results

This section is divided in 2 parts, the first describing the analysis of the EM signals acquired from the microcontroller AT89C51ED (ATMEL, 2007), and the other section focusing on the results obtained from analyzing power traces presented in the DPA book home page WS3 (Mangard et al. 2010a).

In the case of the EM traces obtained from the microcontroller, a number of scenarios which occur in side channel attacks are illustrated. In one case the clock frequency of the processor system changes. A case where the outcome of the analysis clearly returns the correct key is presented. In contrast another analysis returns the correct key as well, but it is not clear if the result is meaningful or not. The effects of adding more traces to the analysis is also explored. The time it takes for each methodology to process the data is also presented. The effects of misalignments in the acquired traces is also explored while the last scenario exemplifies the use of accuracy and estimation.

3.5.1 Analysis of Microcontroller AT89C51ED2

The experimental setup used for this section includes the AT89C51ED2 microcontroller within the Keil MCB251 evaluation board (Keil, 1997). The EM probe was placed over the microcontroller chip. The 1 cm inductive probe (Electro-Metrics Inc., 2004) was connected to a wideband amplifier (also provided by Electrometric) which was connected to the Tektronix TDS7254 digital oscilloscope (Tektronix, 2003). This scope was used for all EM traces captured in this thesis. The trigger for the oscilloscope is controlled using one bit of a parallel port from the microcontroller. The crystal that generates the clock signal for the microcontroller was removed and replaced by a Rohde & Schwarz SMA100A signal generator (R&S, 2011). The main objective of replacing the crystal by a signal generator was to gain control over the clock frequency from the board and set the microcontroller clock frequency to a desired speed with the maximum frequency allowed being 40 MHz. The side channel analyses were performed on 32 sets of 2,048 EM traces experimentally captured with a sampling resolution of 500 MS/s. The attack focuses on the first round of a software implementation of AES written in C running in the microcontroller processing one byte of the key.

After analysing different sets of EM traces using correlation analysis in the frequency domain, it was observed that the maximum correlations for the correct key of the microcontroller occurred in a specific range of frequencies between 30 MHz and 50 MHz. This range of frequencies was found to be independent of the clock speed. Figure 3.1 shows the correlation values for the correct key byte guess when the clock system takes different frequencies between 1 MHz and 40 MHz.

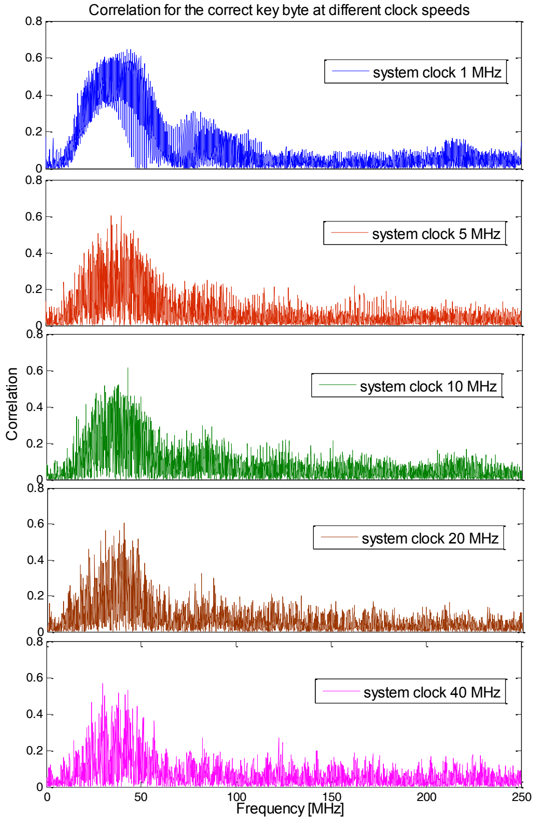


Figure 3.1: Correlations in the frequency domain for the correct key guess using different system clocks.

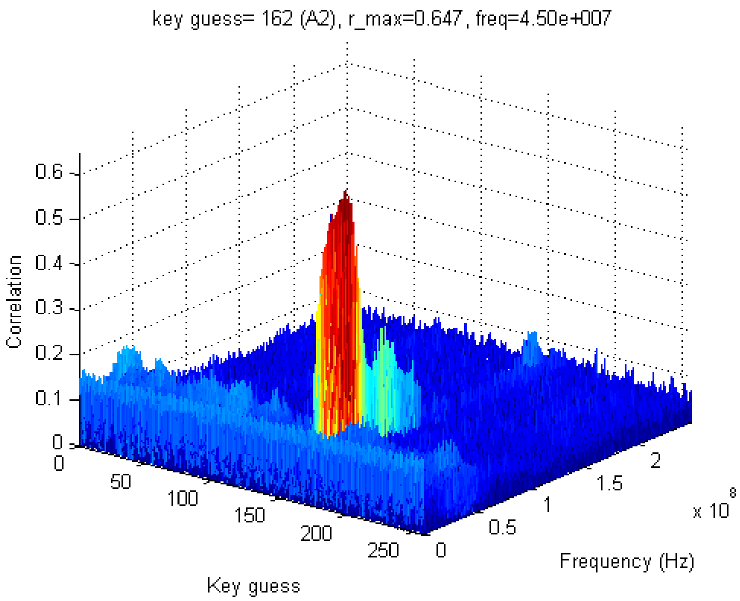


Figure 3.2: Correlation analysis in the frequency domain for 2,048 EM traces using a system clock of 1 MHz.

After setting the clock frequency to 1 MHz, 2,048 traces were captured while the microcontroller was running the AES algorithm using 0xA2 as a small part of the key. Figure 3.2 shows the absolute values for the matrix of correlation values using one set of 2,048 EM traces. In this example, it is clear that the correct key guess is A2. The maximum correlation is clearly larger than most other correlations, in this case the proposed *estimation* metric produced a “1” for δ=3.8.

Sometimes it is not clear how to interpret the outcome of the correlation matrix if the correct key is unknown. Figure 3.3 illustrates an example where despite finding the correct key, it is not clear if the result is meaningful or not since the maximum correlation for the correct key is almost the same size as the other correlations. Here the number of traces used for the analysis was 55. The accuracy metric was 1 while the estimation metric was “0” for δ=3.8.

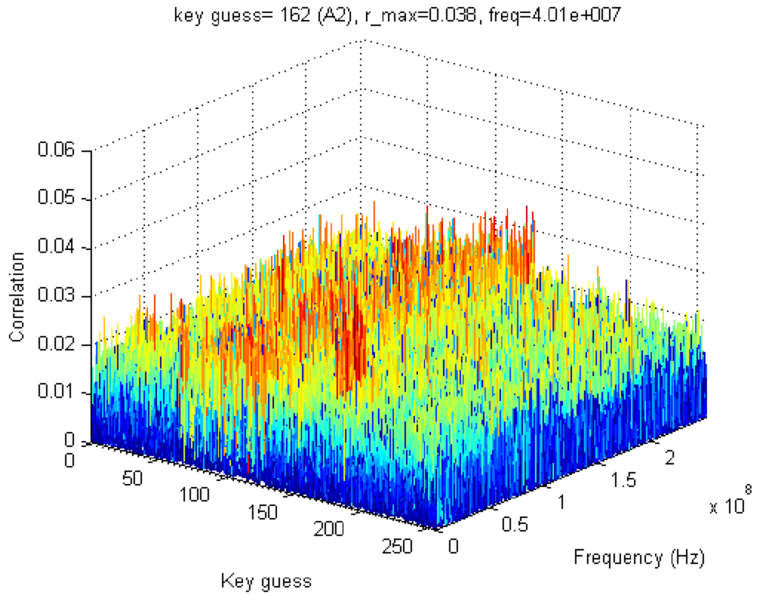


Figure 3.3: Correlation analysis in the frequency domain for 55 EM traces using a system clock of 1 MHz.

The next set of EM traces were obtained when the microcontroller was using a 24 MHz clock. The traces were analyzed using three different correlation methods, correlation analysis in the time domain, correlation analysis in the frequency domain and the proposed fast correlation in the frequency domain. Figure 3.4 compares the results obtained after analyzing from 1 to 100 different traces using each one of the three methods. Although correlation analysis in the time domain returns a correct key guess with only 10 traces, this result is not robust because with one trace more (11 traces), it moves to a wrong key again.

With around 40 traces, the correlation values for the fast correlation in the frequency domain and correlation analysis in the time domain can guess the correct small part of the key.

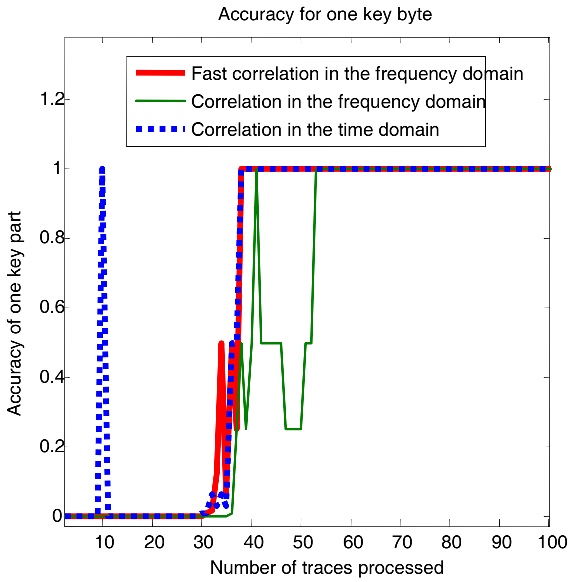


Figure 3.4: Accuracy results after processing EM traces from our microcontroller.

The time required to analyze the acquired traces becomes important when a large number of traces (possibly millions) for a large number of possible key guesses and points of attack need to be analyzed. Figure 3.5 compares the time required by the three different correlation analyses to process one key byte when the microcontroller runs the first round of AES. In this analysis, 1 to 500 EM traces were considered. Each EM trace contained 5,000 samples. In the case of the fast correlation in the frequency domain, the frequencies between 30 MHz and 50 MHz were used and correspond to the frequencies where the magnitudes of the power spectrum are larger. The measured times correspond to the time it takes for a Matlab implementation of the correlation analysis algorithm to run. The figure clearly shows that fast correlation in the frequency domain is the method that requires less processing time followed by correlation analysis in the frequency domain. The slowest was correlation analysis in the time domain.

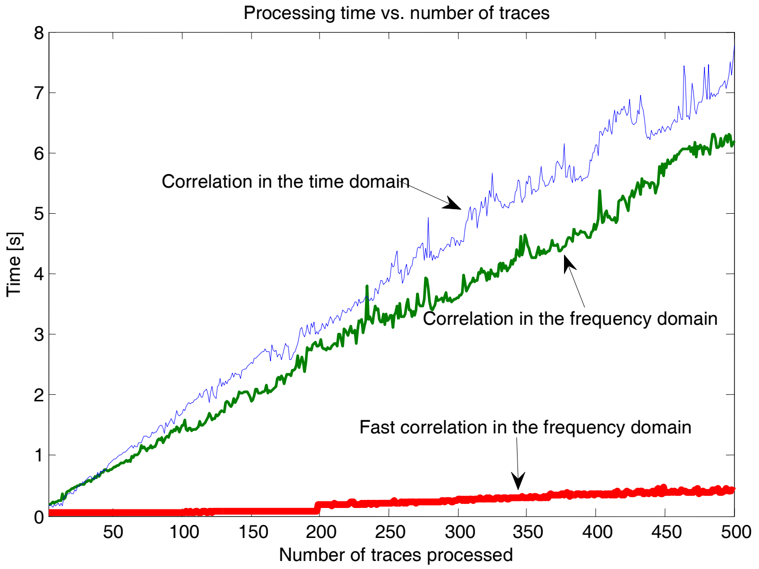


Figure 3.5: Computation time required to process a given number of traces using different correlation methods.

During the process of acquiring EM and power traces, it is not always possible to have a reliable trigger signal attached to the oscilloscope to capture the traces. Thus, trace misalignment is an important issue to consider. In addition, some systems use the insertion of random delays as a countermeasure. For this reason, the effects of random delays on the trace signals were evaluated. Two groups of delays were tested. Delays from 0 ns to 20 ns were randomly inserted into traces in one group. In the other group, delays from 0 ns to 100 ns were randomly inserted. Figure 3.6 presents the result obtained from this analysis. Using correlation analysis in the time domain and random delays from 0 ns to 20 ns, 5 times more traces were required to be able to guess the correct key. For the delays ranking from 0 ns to 100 ns, it was not possible to guess the correct key. For the fast correlation in the frequency domain, the first set of traces with delays from 0 ns to 20 ns had no effect on finding the correct key. For the second set it took only another 10 traces to be able to guess the correct key.

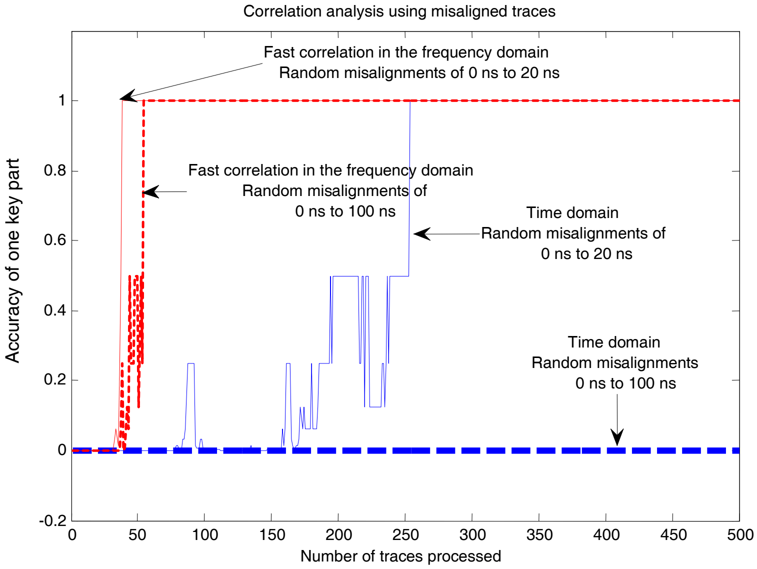


Figure 3.6: Correlation analysis for one key byte using misaligned traces.

In real attacks of electronic devices, sometimes the key is unknown. In such cases, it is very useful to determine the effectiveness of an attack. Figure 3.7 shows the accuracy and estimation values obtained after processing a given number of traces using the fast correlation in the frequency domain. The accuracy metric shows that after processing the first 37 traces, with the fast correlation in the frequency domain the correct key was retrieved. In the figure, the changes in accuracy between 30 and 38 traces are visible. The main reason is that the exponential structure of accuracy helps us to ignore the “noisy” changes, for example, when the correct key changes from position 105 to position 73, and lets us focus on more relevant variations like when the correct key guess changes from the third position to the first position (36 to 37). The estimation metric (using δ=3.8) shows that after processing 55 traces, it is possible to say that independent of the small or large correlation values, the result is meaningful. This binary behaviour of estimation will help us to determine if an attack is meaningful or not when the key is unknown. Changing the value of δ will move the threshold to accept or reject attacks. Further research is necessary to determine the optimal value of δ and its independence from the device under attack.

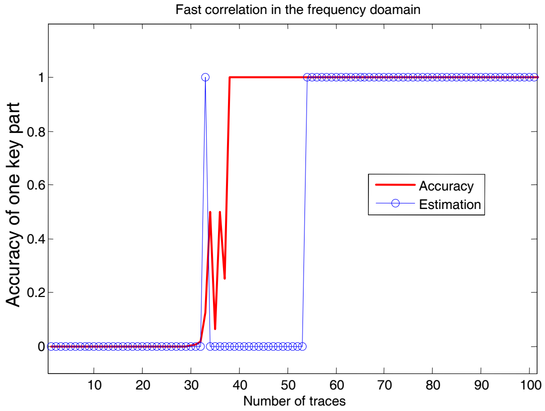


Figure 3.7: Comparison between the estimated effectiveness and the accuracy for different numbers of traces using the fast correlation in the frequency domain.

3.5.2 Analysis of Traces from the DPA book

The power traces from Workspace 3 posted at the DPA book home page in (Mangard et al. 2010a) are analyzed in this section. Previously in (Mangard et al. 2007) a setup is described to perform a power analysis attack on an 8-bit microcontroller that runs at 11 MHz and executes a software version of AES. The microcontroller is mounted on a prototyping board with other basic components. The circuit is powered with a 5 V power supply and a resistor of 1 ohm is connected in series with ground. The oscilloscope probe is connected to the resistor using a differential probe to measure the voltage on the resistor and estimate the power consumption of circuit. The microcontroller communicates with a PC through a RS-232 interface. When the PC sends the data block to the microcontroller for encryption, the microcontroller activates an output on one of its ports previously designed to trigger the oscilloscope and then it starts encrypting the data. For Workspace 3, they measure two sets of the power consumption traces of the circuit when it encrypts 1,000 random plaintexts, one with perfectly aligned traces (without using dummy operations) and one using inserted dummy operations. It uses a software implementation of AES that contains 25,000 power samples, when the microcontroller runs one round of AES, using the 16 byte key.

To analyse the traces using the fast correlation in the frequency domain it was necessary to divide the samples from each trace into a number of windows to isolate the emissions when the microcontroller is processing one part of the key in one round from the emissions of other parts of the key in different rounds. For this 8-bit software implementation, the microcontroller processes in each round 16 small parts of the key. Hence, 16 or more windows should be used. In this case, the 25,000 samples were divided arbitrarily into 40 windows of 625 samples each. This prevents interference between frequencies from one part of the program with others. After using correlation analysis in the frequency domain, the frequencies that returned the highest correlations were identified. From them the ones between 10.5 MHz and 11.5 MHz were used for running fast correlation on the frequency domain. The analyses covered the 40 windows and the ones with the higher correlations for each key byte were selected as the correlation matrix for that key byte guess. In the case of correlation analysis in the time domain the same 40 windows were considered to compare both methods in similar conditions.

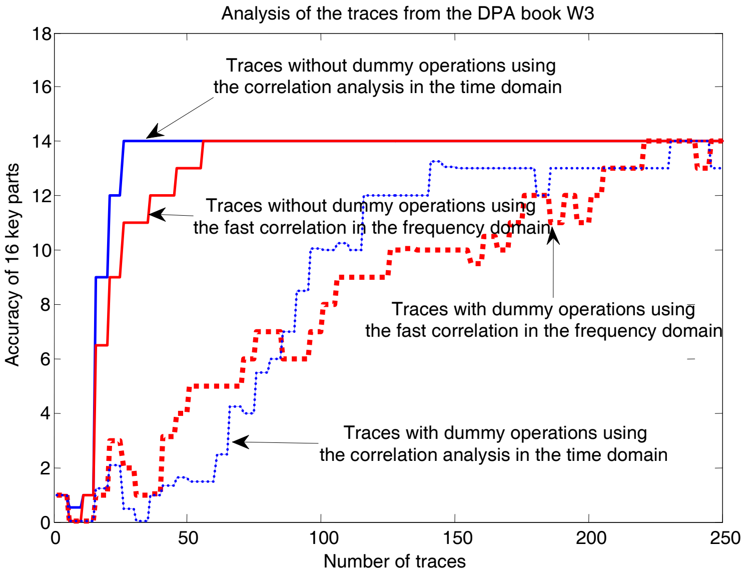


Figure 3.8: Correlation frequency analysis using the traces from the DPA book WS3.

The accuracy for the two sets of traces, one with no dummy operations and the other with dummy operations is compared in Figure 3.8. The effect of the dummy operations on the performance of the analysis is visible. In the case of the traces with dummy operations, the analysis requires more traces to recover the correct key than with no dummy operations. The fast correlation in the frequency domain reaches an accuracy of 14 with fewer traces than correlation analysis in the time domain. It is interesting how the fast correlation in the frequency domain starts with a higher accuracy, then the correlation in the time domain moves ahead but it remains almost constant for close to 100 traces, then the fast correlation in the frequency domain reaches the accuracy of 14 before the time domain. In the case of the traces without dummy operations, correlation analysis in the time domain required fewer traces. From the results obtained, it seems that the traces from Workspace3 only contain the information for 14 of the 16 bytes of the cipher key.

3.6 Comparison to Previous Research and Summary

Unlike previous research that studied correlation analysis in the time domain (Brier et al. 2004, Mangard, 2004, Ors et al. 2004, Kizhvatov, 2009, Standaert et al. 2004), analysis in the frequency domain was also investigated in this chapter (Mateos and Gebotys, 2011). A few other works including (Schimmel et al. 2010, Peng et al. 2009) had studied correlation analysis in the frequency domain; however, they had utilized all frequencies and did not explore the impact of processor clock on the analysis. In this chapter it was shown that, independent of the clock frequency, a small number of frequencies are more likely to leak computing information. Recent research in (Kasper et al. 2011) and (Oswald and Paar, 2011) used correlation analysis in the frequency domain along with real time analog filters (custom hardware) which automatically selects the frequencies to be considered in the analysis.

Previous research had suggested metrics such as the success rate (Standaert et al. 2009), and guessing entropy (Standaert et al. 2009, Kopf and Basin, 2007) but the use of the two proposed metrics, accuracy and estimation, help to show the effectiveness of an attack with a single value instead of multiple values (one for each subkey attacked). In the case of accuracy, it quantifies the correctness of the attack by comparing the key guesses from the attack with the correct cipher key. This is an exponential scale which minimizes the weight from the key guesses ranked away from the top positions. In the case of estimation, it quantifies the effectiveness of the attack without requiring knowledge of the correct key. It estimates the quality of the attack by measuring the distance between the maximum correlation and the mean from all maximums. When it is bigger than a constant δ, the attack is qualified as meaningful.

This chapter uses the EM emissions from the AT89C51ED2 microcontroller as an example to show that a reduced range of frequencies can be associated with the side channel independently of the clock frequency. This fact is the basis of a new methodology called fast correlation in the frequency domain. This type of attack requires the calculation of smaller correlation matrices compared to correlation analysis in the time domain and hence it is faster. It was tested using two different 8-bit processor systems and was able to recover the correct keys using EM emissions and power consumption traces. The results obtained also show fast correlation in the frequency domain is a method immune to misalignments smaller than 20 ns and still able to recover the cipher key without problems even up to misalignments of 100 ns. This characteristic may support viable attacks on devices that use small random delays as a countermeasure. Empirical results show the effectiveness of fast correlation in the frequency domain to recover the keys as a reliable method to retrieve the cryptographic keys. The next chapter presents results from the DPA contest version 2.

Chapter 4

The DPA Contest Version 2

The DPA contest is an event were researchers around the world have the opportunity to compare their side channel attack implementations under equal conditions. It is organized by the VLSI research group from the COMELEC department of the Télécom ParisTech french University (Bulens et al. 2011). The first edition started in 2008 and every year (edition), they switch the contest's topic. The topic for the DPA contest version 2 (DPACv2) (Bulens et al. 2011) was attacking a hardware implementation of AES-128. For the contest, 3 different databases, 2 public and 1 private, were prepared. The first public database contains 1 million traces acquired using 1 million random plaintexts and 1 million random cipher keys. The second public database contains 32 sets of power traces acquired when the circuit was encrypting 20,000 random plaintexts per each one of the 32 cipher keys used. The private database is similar to the second public database, however they used different cipher keys and the traces remain private to compare the attack submissions. Originally, the DPACv2 had a submission deadline of July 14, 2010 and the participants had no knowledge about the performance of other submissions. The organization committee decided to extend the period up to October 31, 2010, and make public the performance from the original period of the attack submissions. This meant that for the extended period, the participants knew beforehand the performance from the first period submissions.

This chapter focuses on the results of an attack submission to the DPACv2, based on the proposed fast correlation analysis in the frequency domain sent in before the original deadline. To avoid bias in the comparisons among attacks only the submissions to the original deadline are considered. An overview of the hardware setup followed by a brief summary of the results of the contest using the private database and a comparative analysis using the public database is presented. Some of the results presented here have been published in (Mateos and Gebotys, 2010) and in the home page of the DPACv2 (Bulens et al. 2011)

4.1 DPA Contest Setup

The setup used for acquiring the power traces analysed in the contest included a hardware version of AES implemented on the SASEBO GII board (RCIS, 2010). This board contains a cryptographic FPGA Xilinx Virtex-5 LX30/LX50 that runs a 128-bit implementation of AES without countermeasures. The board uses a 24 MHz crystal and the circuit runs one round of AES per clock cycle. The power traces were acquired using a digital oscilloscope with a sampling resolution of 5 GS/s and each trace contains 3,253 samples (the 10 rounds of AES in 0.65 µs). From the headers in the trace data files, it is possible to assume each plaintext was encrypted 10 times using the same cipher key and the oscilloscope recorded the average of those 10 traces.

The way in which an interface program, called the "attack wrapper" interacts with the submitted attack program is described in the documentation part of the DPACv2 homepage. In the case of windows based submissions the attack wrapper was implemented using C# and consists of a small application that provides the attacking program with the plaintexts, ciphertexts, and power traces that must be analysed.

The attack wrapper is configured with the number keys that will be analyzed (0 to 31), the number of traces to be used by the attack program, the name of the output file, the directory where the trace files are stored and the round of interest in the attack program. Then, the attack wrapper creates an instance of the class Trace (*trace*) that contains 3 properties: plaintext, ciphertext, and samples. The plaintexts and ciphertexts are arrays of 16 bytes, while "samples" is an array of 6,506 bytes (2 bytes per 3,253 samples). The attack wrapper loads into the object *trace* the values for the first test and passes the *trace* to the attacker. The wrapper records the starting time and waits for a response from the attacker. When it receives a response, it saves the results of the attack, records the ending time and subtracts the starting time from the ending time. This keeps track of the answers and the processing time. Then, the attack wrapper loads the next values into *trace* and passes the object to the attacker for the next test. This cycle continues until the program reaches the number of traces set during the initialization of the attack wrapper.

The responses from the attacker are required to be a matrix of 16 columns by 256 rows where each column corresponds to one byte of the small part of the key associated to one of the s-boxes and every row contains the key guesses sorted from most likely (row 1) to less likely (row 256).

The attack submitted to the DPA contest focused on the last round of AES encryption and a hypothetical power consumption model based on the Hamming weight was selected. Using the traces from the public database and simple power analysis, a region where the last round would most likely occur was determined. This region was found between samples 2,600 and 3,000. Then correlation analysis in the frequency domain was applied to analyze the traces from the public database. A range of select frequencies was determined where bigger magnitudes for the correct key guess were obtained compared to other key guesses. This pre-characterization found that the frequencies between 9 MHz and 20 MHz returned the best results.

4.2 DPA Contest Results

The results of the attack are evaluated using another program called Compute-Results. This program reads the results saved by the attack wrapper. It reads the plaintexts and encrypts them using AES and the cipher key registered in the results file. Compute-Results stores the partial result at the round the attack program is aiming, and searches for the position of the correct values in the responses sent by the attack program. Compute-Results measures the *partial success rate*, *partial guessing entropy*, *global success rate*, and *execution time*. According to (Standaert et al. 2009) the *partial success rate* contains the first-order success rate sampled from 32 experiments and calculated for each of the 16 bytes of the AES subkey. The *partial guessing entropy* uses the guessing entropy as a metric and the *global success rate* measures the first-order success rate in covering the complete key sampled from 32 independent experiments.

Table 4.1 Results from the DPACv2 (Bulens et al. 2011)

Attack	Global Success Rate after 20,000 traces	Minimum-Partial Success Rate after 20,000 traces	Maximum-Partial Guessing Entropy after 20,000 traces	Time/Trace: Mean time per trace
Alexis Bonnecaze, IML, ERISCS, Attack DPA,	0.41	0.75	1.72	< 0.01 s
Alexis Bonnecaze, IML, ERISCS, Attack SPE	0.88	0.94	1.06	0.83 s
Alexis Bonnecaze, IML, ERISCS, Attack VAR	0.53	0.69	9.16	< 0.01 s
Alexis Bonnecaze, IML, ERISCS, Attack VDPA	0.25	0.53	5.22	< 0.01 s
Alexis Bonnecaze, IML, ERISCS, Attack CVM	0.44	0.69	6.56	0.31 s
Antoine Wurcker, UNILIM: Faculte des Sciences et Techniques de Limoges, Attack A	0.81	0.88	1.16	0.25 s
Antoine Wurcker, UNILIM: Faculte des Sciences et Techniques de Limoges, Attack B	0.69	0.88	1.16	0.25 s
Aziz El Aabid, Télécom ParisTech, Template Attack	0.19	0.35	37.84	0.05 s
Edgar Mateos, University of Waterloo, Fast correlation in the frequency domain.	0.59	0.78	3.41	< 0.01 s
Matthieu Walle, Thales Communications, Attack 7F	0.94	0.94	1.09	0.07 s
Matthieu Walle, Thales Communications, Attack 7T	1.00	1.00	1.00	0.03 s
Matthieu Walle, Thales Communications, Attack 9F	0.94	0.94	1.09	0.07 s
Matthieu Walle, Thales Communications, Attack 9T	1.00	1.00	1.00	0.04 s
Maël Berthier, MORPHO, Attack CPA	0.88	0.94	1.06	4.52 s
Sylvain Guilley, Télécom ParisTech, Reference Attack	0.53	0.81	40.25	1.10 s
Thanh-Ha Le, MORPHO, Attack MI cumulant 4th order	0.68	0.82	1.74	8.77 s
Thanh-Ha Le, MORPHO, Attack MI cumulant	0.88	0.91	1.44	7.24 s

The complete report with the results from all attacks submitted to the DPACv2 can be found online at (Bulens et al. 2011). Table 4.1 presents a summary of the results for the attacks submitted in the first (original) period of the contest and is based on the DPACv2 hall of fame results (Bulens et al. 2011). There are four attacks that require less than 0.01 seconds to process each trace, three are written in C++ (DPA, VAR, and VDPA) and one using Matlab (fast correlation in the frequency domain). Among the fastest 4 attacks, the one that uses the proposed fast correlation analysis in the frequency domain obtains the top global success rate; it has the best minimum-partial success rate, and returns the second best maximum partial guessing entropy.

The global success rate for the fast correlation analysis in the frequency domain is shown in Figure 4.1, the success rate starts increasing after 6,000 traces, between 11,000 and 14,000 traces it presents few fluctuations and stops at 0.59 for 20,000 traces. It means that 59% of all small parts of the key were guessed correctly with 20,000 traces.

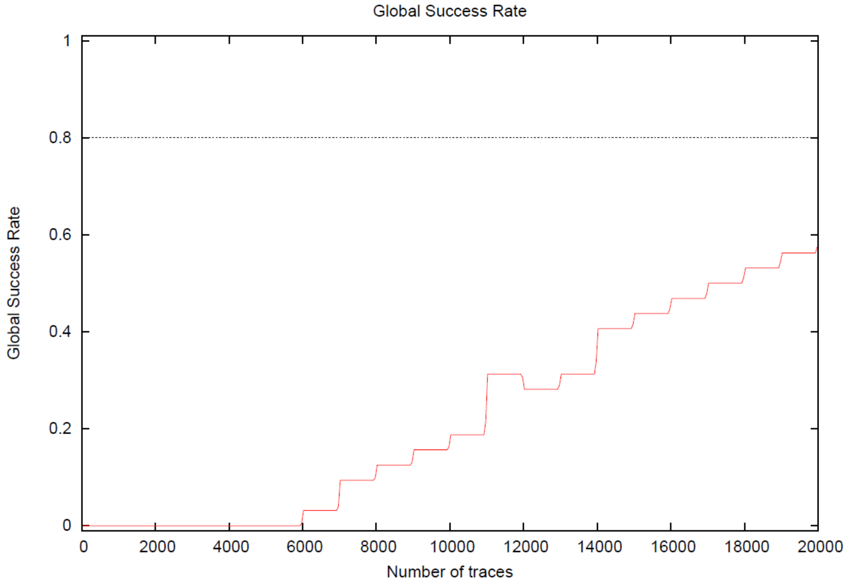


Figure 4.1 Results from the DPACv2 with the global success rate entropy for all the subkeys bytes (Bulens et al. 2011)

The partial success rate reported in the DPACv2 for the fast correlation analysis in the frequency domain attack is shown in Figure 4.2. With 3,000 traces it was possible to obtain the small part of the correct key used in some of the S-boxes. With 20,000 traces only 4 S-boxes did not reach a success rate of 1, being the minimum-partial success rate for 20,000 traces 0.78 (as reported in Table 4.1). The average value for the partial success rate at 20,000 traces is 0.96.

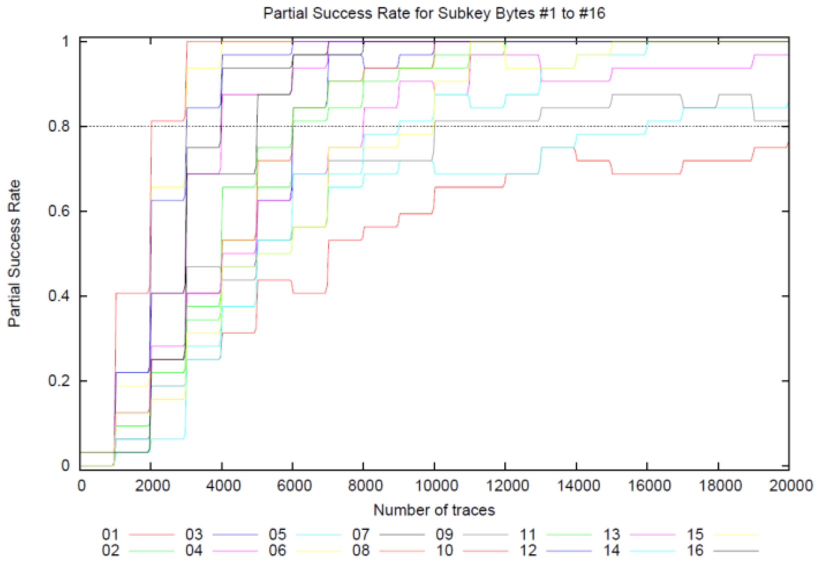


Figure 4.2 Results from the DPACv2 with the partial success rate for all the subkeys bytes (Bulens et al. 2011)

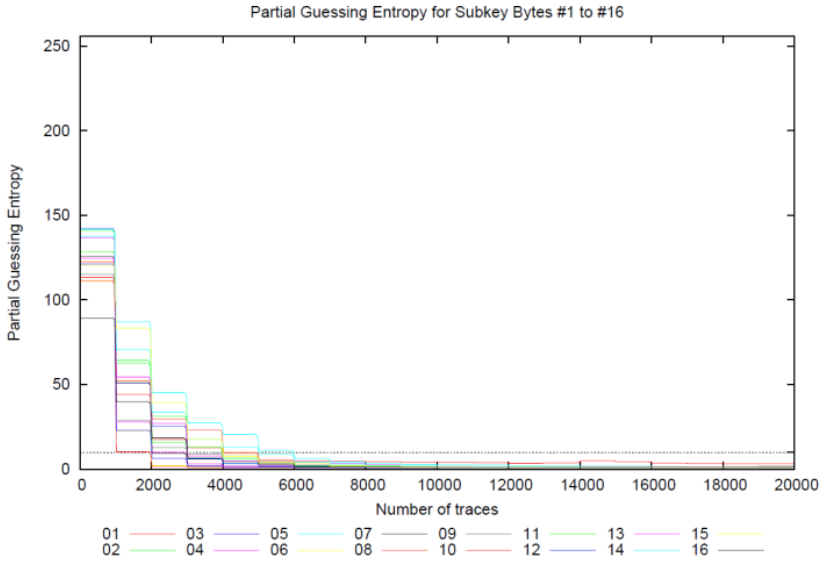


Figure 4.3 Results from the DPACv2 with the partial guessing entropy for all the subkeys bytes (Bulens et al. 2011)

The partial guessing entropy across the number of traces is shown in Figure 4.3. As expected, the guessing entropy starts around 128 (random guess 50-50) and starts decreasing close to 1, where 1 means a correct guess. In Table 4.1 the value reported for the maximum-partial guessing entropy after 20,000 traces using the fast correlation analysis in the frequency domain is 3.41; however, one parameter that could be more representative of the partial guessing entropy is the mean instead of the maximum. The mean of the partial guessing entropy at 20,000 traces is 1.2.

4.3 DPA Contest Public Database

This section uses the traces of the DPACv2 public database to illustrate the performance of two types of correlation analysis discussed in Chapter 3, correlation analysis in the time domain and fast correlation in the frequency domain. Both attacks aim at the last round of AES encryption and for fast correlation in the frequency domain, the analysis takes into consideration the frequencies between 9 MHz and 20 MHz.

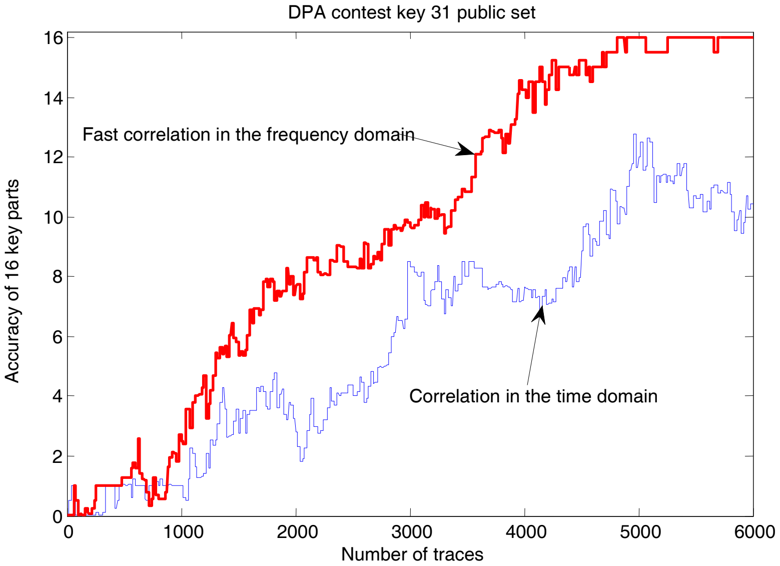


Figure 4.4: Correlation frequency analysis using the traces from the DPA contest public database.

The results of the attacks using fast correlation in the frequency domain and correlation analysis in the time domain are compared in Figure 4.4. The metric applied was accuracy. The traces used correspond to the cipher key number 31 and the figure shows better accuracy for the fast correlation in the frequency domain than the correlation analysis in the time domain. With fast correlation in the frequency domain, it was possible to guess the complete 128 bits of the cipher key using less than 5,000 power traces.

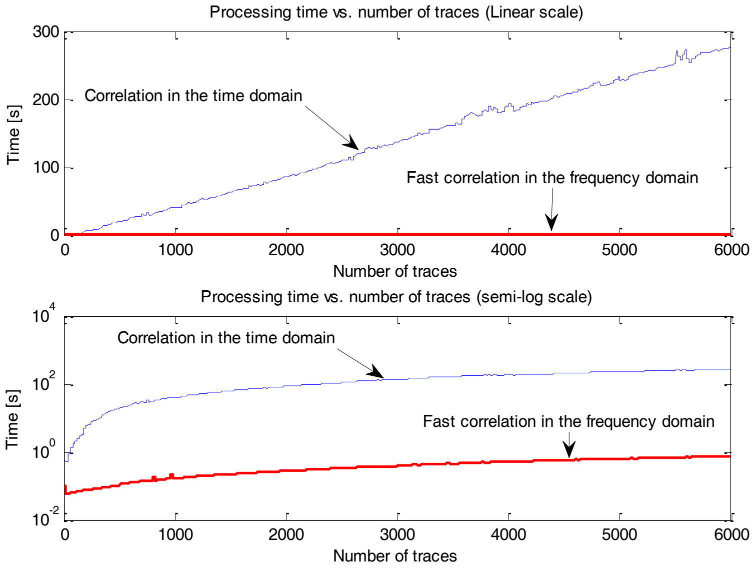


Figure 4.5: Processing time required to compute correlation analysis using traces from the DPA contest public database.

One of the objectives for developing fast correlation in the frequency domain was to reduce the processing time required to analyze a set of power traces. It is interesting to note that one of the latest requirements of the DPACv2 asked for the participants to process 20,000 traces in less than 48 hours. The time to compute each analysis using correlation analysis in the time domain and fast correlation in the frequency domain is compared in Figure 4.5. It shows the computation time to perform correlation analysis using a different number of traces. In the top, a linear scale is presented and in the bottom a semi-log scale is used to read the processing times for fast correlation in the frequency domain. Fast correlation in the frequency domain requires 0.7457 seconds to process 6,000 traces with an accuracy of 16 (finding all the 16 bytes) while using correlation analysis in the time domain, it takes 276.5 seconds to process 6,000 traces with an accuracy of 10.45.

The accuracy results obtained after processing the 32 different keys of the DPA contest public database using fast correlation analysis in the frequency domain are presented in Figure 4.6. The graph shows the evolution from 1 to 10,000 traces with increments of 100 traces. We can observe that after processing 6,000 traces the average accuracy is close to 14 and after 10,000 is close to 15.

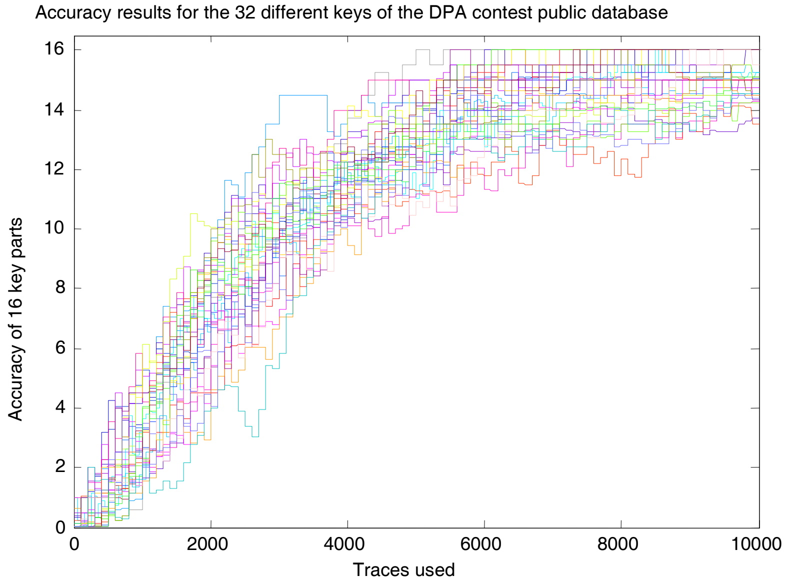


Figure 4.6: Accuracy results using the fast correlation in the frequency domain using power traces from the DPA contest public database.

Sometimes the correct key is unknown and it is necessary to evaluate the effectiveness of an attack. In these cases the use of the metric called *estimation* may help. In the case of the traces from key 31 using δ=3.8, the values obtained for *estimation* and *accuracy* are shown in Figure 4.7. In the figure, the similarity of the two metrics is close enough to forecast what would be the effectiveness of the attack if the correct key were unknown. It is also visible that based upon the value of estimation using 5,000 traces, the attackers could predict that they found the correct key (and in fact they have found the key).

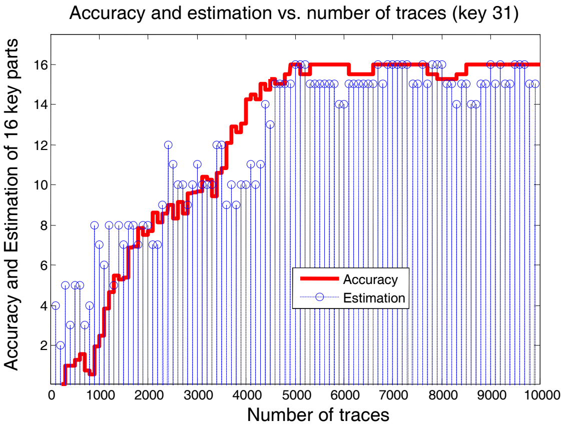


Figure 4.7: Comparison between the estimated effectiveness and the accuracy for different numbers of traces from the DPA contest using fast correlation in the frequency domain.

4.4 Comparison to Previous Research and Summary

A brief summary of the results of the proposed fast correlation analysis submitted to DPACv2 is presented in this chapter. The attack called correlation analysis in the frequency domain was the fastest of all the attacks submitted and among the fastest attacks, it had the best success rate at 20,000 traces. It was able to retrieve a number of 128-bit cipher keys using traces from the public and private database. Hence, using a reduced range of frequencies, it is possible to attack a 128-bit implementation of AES without countermeasures when it runs in the cryptographic Xilinx Virtex-5 LX30/LX50 FPGA. The most accurate analysis in the competition used on average 0.03 seconds while our technique required less than 0.01 seconds (the minimum discriminate in the measured time was 0.01 s. Comparing the processing time between fast correlation in the frequency domain and correlation analysis in the time domain, the first one is approximately 370 times faster and has better accuracy. Additionally, an example is presented where using the estimation metric the effectiveness of an attack could be forecast without knowing the correct key guess. An alternative EM probe that uses giant magnetoresistors is described in the next chapter.

Chapter 5

Side Channel Analysis using Giant Magneto-Resistive Sensors

This chapter explores the use of Giant Magneto-Resistive (GMR) sensors in the field of side channel analysis. First, a brief introduction about the Giant Magneto-Resistive phenomenon is presented. Then, using a commercial EM inductive probe of 1 cm in diameter and a GMR, a number of EM traces are acquired and analysed. These traces were obtained from the 8-bit microcontroller AT89C51ED when it was running a software version of AES. As mentioned in Chapter 2, the use of this microcontroller is of particular interest for this research considering the JCOP30 and JCOP41 Java Cards are based on extended architectures of the 8051's microcontroller (NXP, 2009c). For the evaluation of the GMR probe, four different sampling rates were used ranging from 500 MS/s to 50 MS/s and the results were compared with those from a commercial probe with performance that is well documented (Electro-Metrics Inc., 2004) and has been effective in the search of side channel analysis (Mateos and Gebotys, 2010), (Gebotys, 2006), (Gebotys et al. 2005). The acquired traces are analysed in the time and frequency domain. Most of the work presented in this chapter has been presented in (Mateos and Gebotys, 2011).

5.1 Introduction to GMR

The use of electromagnetic emissions to recover cryptographic information has been extensively exploited, such as research in (Messerges et al. 2002, Oswald and Paar 2011, Quisquater and Samyde 2001, Gandolfi et al. 2001, Agrawal et al. 2002, Carluccio et al. 2005, Mateos and Gebotys 2010, Thanh-Ha Le et al. 2006, Kizhvatov, 2009, Kasper et al. 2009, Le et al. 2008, Peng et al. 2009, Gebotys et al. 2005, Aerts et al. 2006, Chen et al. 2008, Kasper et al. 2011, Mangard, 2003, Plos et al. 2008, Yamaguchi et al. 2010). The effects of EM sensors in side channel analysis have been studied in only a limited number of these works. In (Agrawal et al. 2002) they found that the most effective near field probes were those made of a small plate of a highly conducting metal like silver or copper attached to a coaxial cable. In (Gandolfi et al. 2001) different types of sensors were studied including integrated inductors and magnetic loops, but the best EM signals were collected using handmade coils made of copper whose diameter varied from 150 microns to 500 microns. Research in (Quisquater and Samyde, 2001) used flat coils positioned with a motorized table that set the sensor with micrometric precision. Other researchers (Aerts et al. 2006) studied different topologies of EM probes trying to identify the shape that delivers the best results. Others (Yamaguchi et al. 2010) implemented an inductive square shape probe of 180 μm by 180 μm in a LSI circuit. Most of the work done in EM side channel analysis focused on the use of inductive probes and the use of the giant magnetoresistive (GMR) effect has not been investigated for side channel analysis.

Lord Kelvin documented the magnetoresistance phenomenon in 1857. The giant magnetoresistance was discovered by Peter Grünberg and Albert Fert in the late 1980s and is widely used in the read head of hard disc drives (Kasap, 2006), (Mallinson, 2002). The word "giant" refers to the large change in resistance that occurs in these devices (10% to 20%) when they are in the presence of a magnetic field. The basic structure of these devices is two ferromagnetic metal films (such as Fe or Co or their alloys, etc.) separated by a metallic nonmagnetic film (such as Cu). The magnetic layers are thin (less than 10 nm) and the nonmagnetic layer is thinner (Kasap, 2006).

In the absence of external magnetic fields, the magnetic moment of the layers adjacent to the copper face different directions, due to the antiferromagnetic coupling of the built device. Normally, copper is a good conductor; however, when it is a few atoms thick, electron scattering increases its resistance notoriously. This resistance depends on the relative orientation of the electron spins next to the thin copper layer. When an external magnetic field is applied and the magnetic moments of the adjacent layers are aligned in the same direction, the resistance decreases (NVE).

5.2 Preliminary Results using Giant Magneto-Resistive Sensors

The AT89C51ED2 microcontroller within the Keil MCB251 evaluation board was used to analyze the performance of the GMR and inductive probes. The board uses a 24 MHz crystal, thus in consideration of the Nyquist-Shannon sampling theorem, the EM traces were captured using different resolutions above 2 times the clock frequency. The resolutions analyzed include 500 MS/s, 250 MS/s, 125 MS/s, and 50 MS/s. The probes used were the commercial 1 cm loop EM probe and the GMR probe. The GMR probe uses the NVA AB001-02 sensor (NVE). This sensor has two pairs of unshielded resistors in a Wheatstone bridge configuration. It comes in a surface mounted, 8 pin package (MSOP8) and one pair of the resistors are located on the area among the pins 1, 2, 7, and 8 while the other is in the area among the pins 3, 4, 5, and 6 (NVE). The point between pins 1, 2, 7, and 8 is referred in this research as the position point of the GMR probe. The commercial inductive probe was placed on top of the microcontroller and was manually moved searching for a point where the magnitude of the EM signals increased while the microcontroller was continuously writing and reading from the memory. The GMR probe was positioned in the same area as the inductive probe and the axis of sensitivity was shifted to form a 45°angle with respect to the edges of the microcontroller. In this way, the sensor was able to detect 70.7% of the magnetic field coming from the X and Y axis. For example, placing the sensor parallel to one of the axes let say X, it would ignore the magnetic field coming from the opposite axis (Y). The attack focus was the microcontroller processing one byte of the key during the first round of AES. For the cases of 500 MS/s, 250 MS/s, and 125 MS/s, the results were verified utilizing 10 sets of 2,048 traces each. For the sampling rate of 50 MS/s, 50 sets of 2,048 traces were used. The evaluation board with the GMR probe is shown in Figure 5.1.

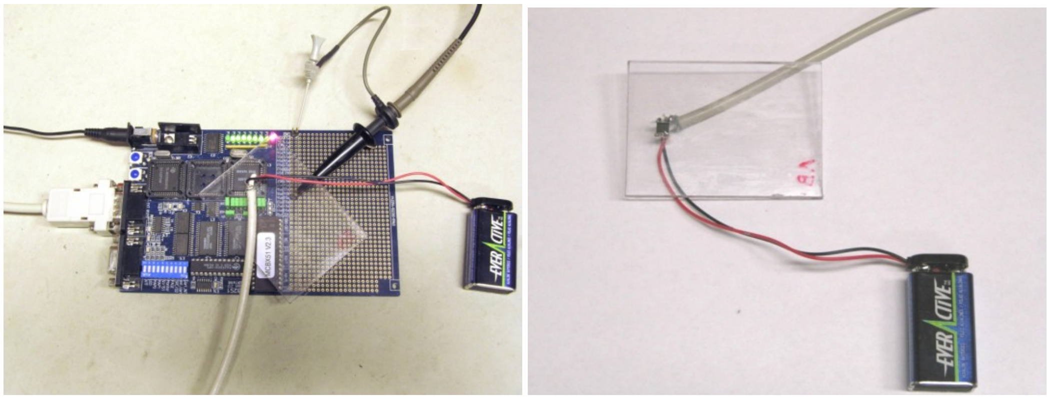


Figure 5.1: Picture of the evaluation board used showing the position of the GMR probe (left). Close up of the GMR probe (right)

5.2.1 Correlation Analysis in the Time Domain

In the case of correlation analysis in the time domain, different sets of traces captured using the inductive probe (EM Probe) and the GRM probe were analysed. Table 5.1 presents the maximum values of the correlation matrix when the EM traces obtained with the inductive and GMR probe are analysed using correlation analysis in the time domain. In the table, most of the correlation values for the inductive probe are larger than those for the GMR probe. However, in the case of the GMR probe, the correct key was recovered in all the cases, even where the inductive probe failed in 39 of 50 experiments with a sampling resolution of 50 MS/s. The analysis in the time domain was successful in recovering the correct key in all the cases for both probes using resolutions of 500 MS/s, 250 MS/s and 125 MS/s.

Table 5.1: Maximum correlation for the time domain analysis

Correlation coefficients in the time domain		
Sampling frequency [MS/s]	EM probe	GMR probe
500	0.913	0.523
250	0.708	0.426
125	0.629	0.306
50	0.161*	0.212

* Able to recover the correct key in only 11 of 50 experiments.

Next, correlation values for the extreme cases of 500 MS/s and 50 MS/s were explored. Figure 5.2 shows the values of the correlation matrix for the case of 500 MS/s, using the EM probe and correlation analysis in the time domain. In this case, a clear spike appears at 7.35 μs for the key guess 162. It is evident that it corresponds to the correct value of the key.

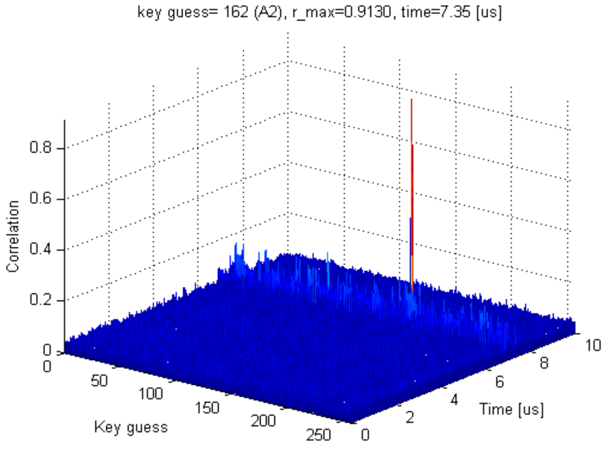


Figure 5.2: Correlation matrix in the time domain for 2,048 traces using EM probe and a sampling frequency of 500 MS/s.

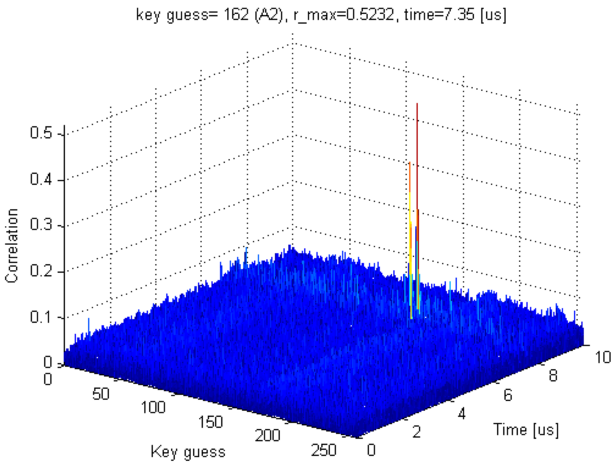


Figure 5.3: Correlation matrix in the time domain for 2,048 traces using the GMR probe and a sampling frequency of 500 MS/s.

The maximum correlation for the GMR probe using correlation analysis in the time domain is shown in Figure 5.3. Similar to the inductive probe the maximum correlation of 0.523 occurs at 7.35 μs. However, with the GMR probe a second spike is visible for the same correct key guess (162) at 7.032 μs.

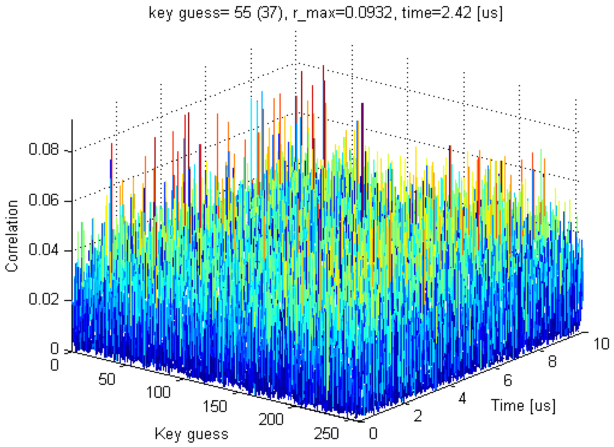


Figure 5.4: Correlation matrix in the time domain for 2,048 traces using EM probe and a sampling frequency of 50 MS/s. The analysis returns a wrong key.

The correlation results after using 2,048 traces acquired using an inductive probe and a sampling resolution of 50 MS/s are presented in Figure 5.4. The key guessed is 55 with a maximum correlation of 0.0932 at 2.42 μs. In this case, correlation analysis in the time domain using the inductive probe was unable to recover the correct key (162).

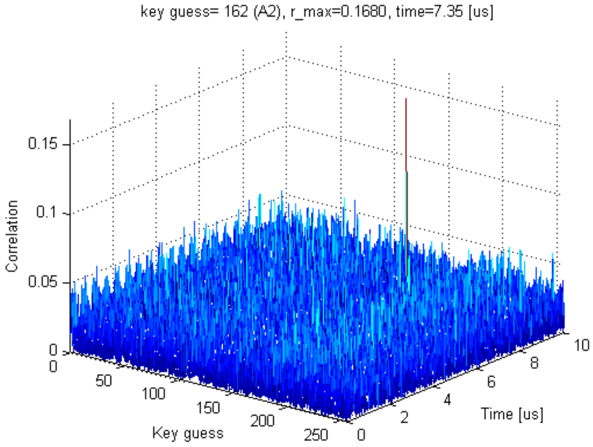


Figure 5.5: Correlation matrix in the time domain for 2,048 traces using the GMR probe and a sampling frequency of 50 MS/s.

For the case of the 50 MS/s using the GMR probe, Figure 5.5 shows the correct key guess (162) with a spike at 7.35 μs. As expected it is located at the same time where other analyses with higher resolutions revealed the correct key. It is important to emphasize the clarity of this result considering the proximity to the minimum sampling rate according to the Nyquist-Shannon sampling theorem. For this scenario, 50 sets of 2,048 traces were captured. In the 50 experiments, the GMR probe was able to guess the correct key at all the times. In the same scenario, the inductive probe failed to recover the correct key in 39 of the 50 experiments.

5.2.2 Correlation Analysis in the Frequency Domain

For correlation analysis in the frequency domain, the same EM traces were analysed as in the case of the time domain analysis. Table 5.2 presents the maximum values from the correlation matrix when the traces were analysed in the frequency domain. The respective correlation values in the table are all smaller than the values obtained in the time domain.

Table 5.2: Maximum correlation values for the frequency domain analysis

Correlation coefficients in the frequency domain		
Sampling frequency [MS/s]	EM probe	GMR probe
500	0.680	0.370
250	0.435	0.320
125	0.231	0.176
50	0.132*	0.110*

* Able to recover the correct key in only 7 of 50 experiments.

For the sampling rates of 500 MS/s, 250 MS/s and 125 MS/s it was possible to recover the correct key using both probes. For the sampling rate of 50 MS/s the GMR and the inductive probe were able to recover the correct key in 7 of 50 experiments.

The correlation matrix plot for the inductive probe using a sampling rate of 500 MS/s is shown in Figure 5.6. It shows that the frequencies between 15 and 70 MHz present the higher correlations for the correct key guess, where the maximum occurs at 41.52 MHz.

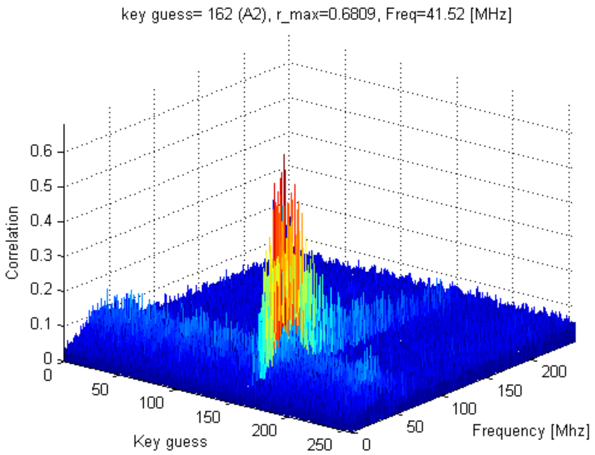


Figure 5.6: Correlation matrix in the frequency domain for 2,048 traces using EM probe and a sampling frequency of 500 MS/s.

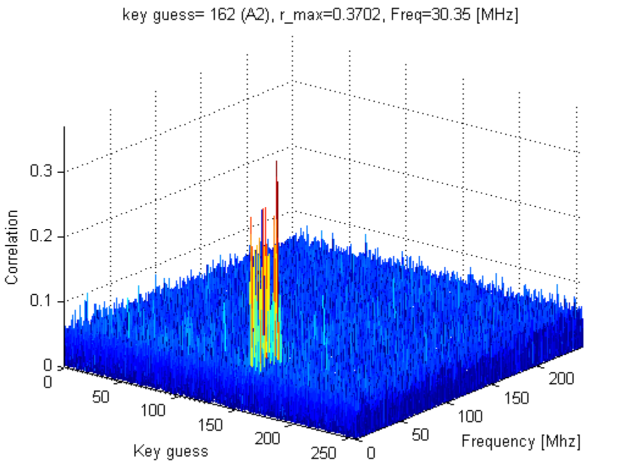


Figure 5.7: Correlation matrix in the frequency domain for 2,048 traces using GMR probe and a sampling frequency of 500 MS/s.

The values obtained after analyzing the traces obtained with the GMR probe, with a sampling rate of 500 MS/s are presented in Figure 5.7. In this case, the correct key was retrieved and the range of frequencies with higher correlations for the correct key are between 1 MHz and 33 MHz.

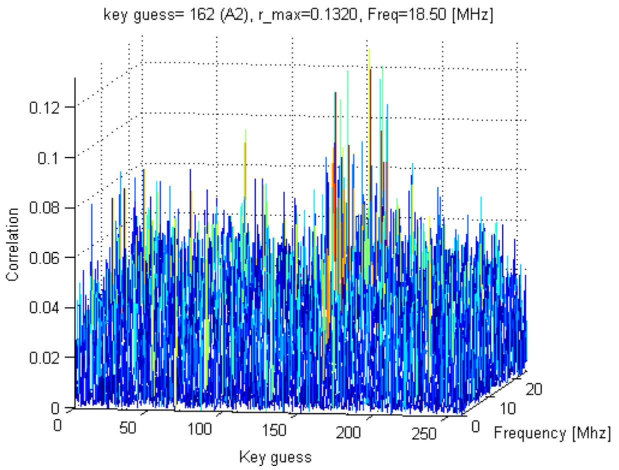


Figure 5.8: Correlation matrix in the frequency domain for 2,048 traces using EM probe and a sampling frequency of 50 MS/s.

The results obtained after analyzing the EM traces with the inductive probe captured using a sampling frequency of 50 MS/s are shown in Figure 5.8. The figure shows 1 result of the 7 cases where it was possible to guess the correct key. Here the maximum correlation appeared at 18.5 MHz.

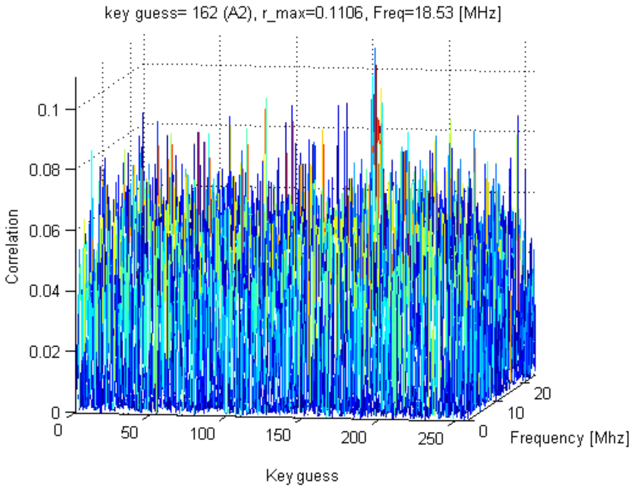


Figure 5.9: Correlation matrix in the frequency domain for 2,048 traces using GMR probe and a sampling frequency of 50 MS/s.

The correlation values for the frequency domain analysis using the GMR traces acquired with 50 MS/s resolution is represented in Figure 5.9. The figure shows the maximum correlation in the correlation matrix with a magnitude of 0.11 and corresponds to the frequency of 18.53 MHz. In this analysis, similar to the EM probe, only 7 of 50 experiments were able to recover the correct key.

5.3 Comparison to Previous Research and Summary

Previous research has studied the effects of different sensors in side channel analysis (Quisquater and Samyde 2001, Gandolfi et al. 2001, Agrawal et al. 2002, Aerts et al. 2006, Chen et al. 2008, Yamaguchi et al. 2010), but all these works utilized only inductive probes of different sizes and materials. The giant magnetoresistance GMR effect has not been previously investigated for side channel analysis. A comparative analysis between an inductive EM probe and a new type of probe that uses a giant magnetoresistance (GMR) sensor was presented in this chapter. The results show successful attacks on an 8-bit software implementation of AES using the GMR probe in some conditions where the inductive EM probe fails to return the correct key. The analysis of the traces obtained with the GMR probe were able to recover the correct key in the time domain for all cases while in the case of the inductive probe using 50 MS/s it recovers the correct key in only 11 out of 50 experiments. Although the correlation analysis for the traces acquired using an inductive probe show higher magnitudes, the axis of sensitivity of the GMR probe was rotated 45° with respect to the edges of the microcontroller and the sensor was detecting 70.7% of the magnetic field.

Using correlation analysis in the frequency domain, it was possible to recover the correct key when the traces acquired had a sampling resolution of 500 MS/s, 250 MS/s and 125 MS/s independently of the type of probe used. In the case of the 50 MS/s both probes succeeded in 7 out of 50 cases. In all the cases tested, the traces with higher resolution returned bigger correlations. However, to test the boundaries of the GMR probe, low resolutions were considered as well. The results presented in this chapter do not imply the replacement of inductive EM probes by GMR probes, but further research on the effects of some properties of the side channel analysis such as their ability to react to magnetic fields in some directions and their response gradient of magnetic fields is recommended. In the next chapter the setup used to study possible side channel from the Java Cards is presented.

Chapter 6

Experimental Setup for Java Card Analysis

In this section, the main experimental setup used to investigate the security of the JCOP30 and JCOP41 Java Cards when they are working in contactless mode and running a Java cryptographic algorithm is described. Acquiring the proper set of power or EM power traces is fundamental yet challenging for side channel analysis of real embedded systems. The proposed setup aims to improve the quality of the EM traces acquired for analysis. In Chapter 5, a new EM probe was presented that is used in this study; however, there are other factors that also affect the characteristics of the acquired traces. These factors include the position of the EM probe, the program implementation used, the sampling rate applied, etc. In this chapter, the modifications made to the commercial card reader are explained. The electronic circuit developed to trigger the oscilloscope using the signals coming from a commercial smartcard reader is also described. Additionally, the programs used in the card and some characteristics of the oscilloscope used to acquire traces are briefly illustrated.

6.1 Card Reader Modifications

The quality of an EM trace is fundamental to side channel analysis. In Chapter 3, the EM principles which form the basis of EM power analysis were reviewed. One factor that determines the magnitude of the EM field is the distance between the emitter and the receptor. Consequently, the position of the probe is essential. For this reason, the first modification to the card reader is intended to assist in the placement of the EM probe. The second change improves the quality of the card reader's power supply by changing its connection from the computer USB port to a lab power supply. The third change, consists of replacing the card reader's internal clock (a surface mount IC) with a more robust and stable signal generator.

6.1.1 The Axis Table.

One of the objectives of this research is to explore the security of the smart cards when they are working in contactless mode. Two aspects to consider are the position of the smart card with respect to the card reader and the position of the EM probe. Previous research has pointed out the importance of setting the position of the sensor with micrometric precision and worked on a motorized table that moved the sensor above the chip using stepper engines to control the position of the screws (Quisquater and Samyde, 2001). In other work, (Kresalek et al. 2008) developed a semiautomatic measurement system to determine the radiating parts of an electronic device. In this case, an XY plotter controlled with a programmable DC power supply is used to position their 2 mm near-field probe. With their setup the maximum resolution to place the probe is 0.02 mm.

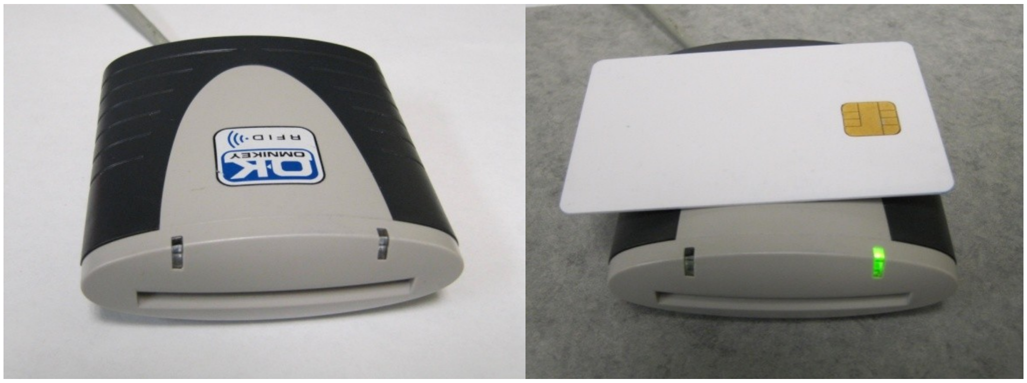


Figure 6.1: CardMan 5121, dual interface card reader

The commercial card reader used in this study is the Omnikey CardMan 5121 (OMNIKEY, 2005). This device has a curved shape cover that makes it difficult to hold smart cards at the same point all the time (see Figure 6.1). To overcome this situation the card reader was removed from its plastic case and placed onto a coordinates table. The board has a space to hold the card reader in such a way that the smart cards can be placed on top of the card reader with minimal variation. In addition, it has two rails that help to slide a connection grid in the X-axis. The connection grid is an interface that helps combine the movement of the sensor in the X and Y directions. It consists of a “number sign” shape made by a pair of X-axis rails fixed orthogonally to another pair of Y-axis rails. On top of this connection grid, the base of the sensor, which has two Y-axis rails that support moving the sensor in the Y-axis, is placed. Figure 6.2 shows the way the 3 main sections interconnect.

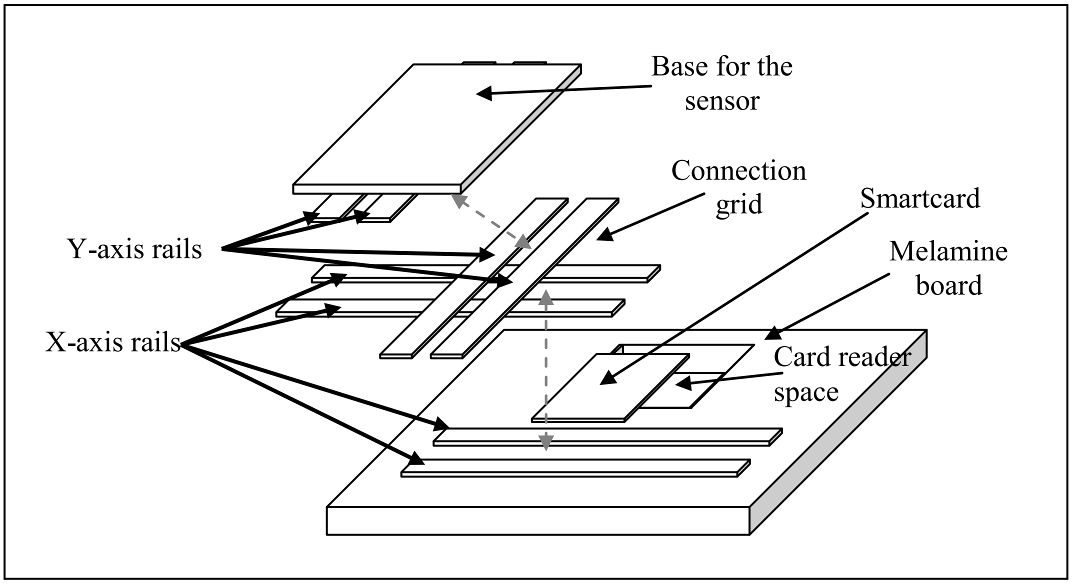


Figure 6.2: Layout of the axis table developed. (Draw not to scale)

To reduce the interference with the electric and magnetic field of the card reader and smart card, the axis table was built avoiding metallic materials. Melamine was selected for the board and wood for the rails. Figure 6.3 illustrates the home made axis table. This setup made it possible to manually place the sensor and the card in specific points and record their positions using a calliper with a precision of ±0.1 mm.

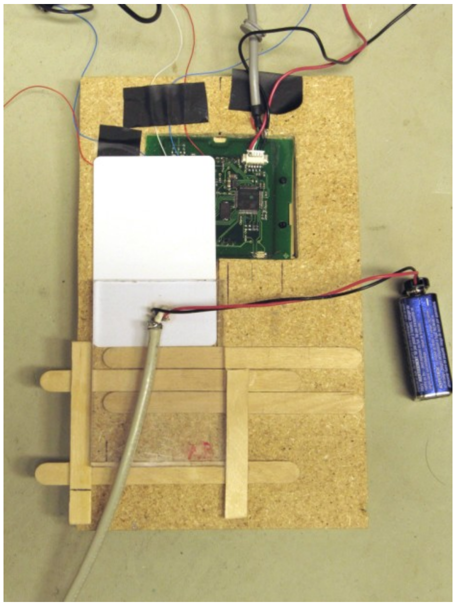


Figure 6.3: Axis table developed to place the smart card and EM sensor.

6.1.2 Power Supplies

The card reader uses the Vcc connection from the USB port to supply energy to all its internal circuits. One of these circuits is the Multiple Protocol Contactless Reader IC (CL RC632) (NXP, 2009b). It modulates and demodulates all communication between the smart card and the card reader. One section of this circuit corresponds to the transmitter control which is connected to a couple of buffers that deliver the modulated 13.56 MHz energy carrier to the antenna. These buffers require a transmitter power supply. In the case of the card reader, this pin is connected to the Vcc from the USB port.

The Vcc from the USB port is connected to the power supply of the computer and despite all its filters, the output voltage contains a number of harmonics having a maximum variation measured with the oscilloscope of 960 mV peak to peak. These values exceed the ±5% tolerance established by the IEC 61967-1 standard (IEC, 2002). This standard describes the conditions for measurement of conducted and radiated electromagnetic disturbances on integrated circuits. To reduce the variability and avoid these harmonics that distort the carrier's waveform, the card reader was connected to an external power supply instead of the USB port. Hewlett Packard power supply models

6235A and 6323A were tested, and the model 6235A was selected for the experiments since it offered the best results by reducing the noise from 960 mV to less than 80 mV peak to peak.

6.1.3 Signal Generator

The Multiple Protocol Contactless Reader IC (CL RC632) is the circuit used by the card reader to communicate with the smart card in contactless mode. This circuit requires a 13.56 MHz oscillator. In the case of the card reader used for this research, the oscillator used is the IC surface mount LIM-T 13560 KDK 5e. This crystal has a frequency stability of ±50 ppm (0.005%=3.7 ps). After detecting the trigger sequence, it is necessary to hold the oscilloscope trigger approximately 1.5 ms before starting the acquisitions of the EM power traces. Using the oscilloscope to measure the variation time, the measurements indicate an average misalignment of ±3 ns.

To improve the quality of the card reader's carrier, the crystal oscillator was replaced with a 13.56 MHz sine signal generated with a R&S SMA 100A signal generator. The resulting signal has a jitter smaller than 19 fs when working at 13.56 MHz.

6.2 Triggering the Oscilloscope

Triggering the oscilloscope at the correct time is one of the most important elements for acquiring meaningful power and EM power traces for side channel analysis. A proper trigger helps to acquire the intended traces and to reduce misalignments among them. In the case of smart cards, they do not have physical connections to trigger a signal for the oscilloscope. In (Mangard, 2003), the author could not find a trigger for the oscilloscope based on the far field radiated emissions. Instead, they used a trigger that was connected physically with the card reader because they found it "impossible" to locate a trigger for the oscilloscope based on the radiated emissions.

6.2.1 Using the Pause at the Start of a Command Transmission

One approach to trigger the oscilloscope is using the "pause" at the start of a command transmission. As described in Section 2.3.1, the Java Cards analyzed use the ISO 14443-2 standard and specifically the type A communication protocol that uses ASK 100%

modulation. A pause is when the carrier's voltage reduces to less than 5% of its initial value for an interval between 0.5 μs to 3 μs. For triggering the oscilloscope, the voltage of the carrier is monitored continuously and when the oscilloscope detects a pause, a trigger event is generated. The oscilloscope will not accept another trigger until the card finishes receiving the command, processing the data, and transmitting a response to the card reader. A similar approach was used in (Berkes, 2008) to trigger the instrument.

This implementation is simple and only requires programming the trigger event in the oscilloscope. One disadvantage of this approach is that the oscilloscope will trigger at the beginning of any command coming from the card reader independently of whether it is an idle command or any other command. Thus, this issue creates uncertainty about which command generated the trigger.

6.2.2 Using a Customized Circuit for Triggering

A more robust approach for triggering the oscilloscope consists of starting the acquisitions after a programmed sequence of commands occurs. For this purpose an electronic circuit that decodes the transmission commands and decides when to trigger or not depending on the present command and previous trigger events was developed. This design prevents triggering on out of sequence or unexpected plaintext/ciphertext (including idle commands).

The ASK decoder designed contains two main modules; a reader decoder that demodulates the card reader carrier and produces 9-bit words; and a 32-bit microcontroller that reads the 9-bit words and runs a C program to determine when and if to trigger. The circuit was implemented in an Altera DE2 board using the Cyclone II FPGA. The reader decoder is based on a number of frequency dividers and the trigger controller uses a 32-bit microcontroller based on the NIOS II CPU.

6.3 Oscilloscope Acquisition Modes

The oscilloscope setup supports the capture of multiple traces and their storage in memory for future processing (Tektronix, 2003). One trace is the set of signal samples captured by one channel of the oscilloscope after it triggers once. A frame refers to the set of traces captured by the active channels and it is possible to have more than one trace in one frame; however, it is only possible to store one single channel at a time in a file hence each frame stored contains one trace. The oscilloscope has a function called FastFrame^{TM} which in a single sequence acquisition allows the capture of *x* number of frames after their respective triggers. These frames can be later stored as a single file and are referred to in this research as file acquisitions. The physical memory of the oscilloscope supports the acquisition of a maximum of 32,000,000 sample points, which can be distributed in 1 trace of 32 M samples, or 2 of 16 M samples, all the way up to 64,000 traces of 500 samples. The maximum captured time depends on the resolution used. In this research, primarily resolutions of 500 MS/s and 250 MS/s were used.

The oscilloscope has three main modes of acquiring traces: sample mode, peak detection mode, and high-resolution mode. In sample mode, the oscilloscope does not post-process the acquired traces. In peak detection mode, it alternates between saving the lowest and the highest value every two acquisition intervals, where an acquisition interval refers to the waveform duration divided by the record length. For high-resolution mode, the instrument calculates the average of all samples taken during an acquisition interval. All acquisitions used in this research were taken using sampling mode unless otherwise stated.

6.4 Applet Used for Analysis

As explained in Section 2.4.1 and later used in Chapters 3, 4, and 5, the implementations of AES are typically focused on the attack of the S-boxes. In attacks presented in Chapters 3 and 5, the AES implementation is attacked in the first round while in Chapter 4, an attack on the last round is presented. When the attack occurs in round 1, the intermediate result attacked is a function of 1 byte of plaintext exclusive-ored by 1 byte of the secret key.

When the electronic device under study uses a precharged bus, the hypothetical power model after the SubBytes transformations corresponds to the Hamming weight of the S-box result (Brier et al. 2004). In the case of a nonprecharged bus, the previous state of the variable or register would be required to calculate the Hamming distance of the S-box to estimate the power model. In the case of the smart cards used, the information about the type of bus used was not available. Thus, to improve the likelihood of an attack, the running program writes a value of 0x00 into a variable and next it writes the result of the S-box(plaintext ⊕ key) into the same variable.

6.4.1 Timing and Resources Constraints

To launch an attack, it is first necessary to determine the times when the card is processing the cryptographic information. According to the data sheets of the Java Cards under study (NXP, 2009a), it is likely that they use some undisclosed DPA countermeasures. Using simple power analysis and differential power analysis, some points of interest (POI) that indicated the time at which AES was entering the S-boxes as mentioned in (Baer et al. 2010) were analysed but did not produce any connection to the keys.

The JCOP30 Java Card does not contain a hardware implementation of AES (Philips, 2003) and while the specification sheet from the JCOP41 (Philips, 2006) refers to an AES coprocessor, a Java Card forum hosted by Oracle points out that the hardware version of AES was accidentally switched off during the production (Anonymous, 2009) and cannot be re-enabled (Svenda, 2009). The unoptimized software implementation of AES that was prepared takes approximately 3 seconds for enciphering a 128-bit plaintext using JCOP30. This means that using the whole 32,000,000 sample points from the oscilloscope and a sampling resolution of 500 MS/s, it would be possible to capture only 6.4% of the full encryption time. To delimit a possible time of attack, a Java applet that runs only the first SubByte transformation from the first round was developed. The smart card receives a plaintext and using a small part of the key, it writes into a variable V the result of the SubByte transformation from the plaintext exclusive-ored with the small part of the key. Algorithm 6.1 describes how applet test 1 works. Step 5 of the algorithm sends the value V to the card reader and verifies what value was received and processed by the smart card. In this way, one detects when the smart card misses a command and one reduces the risk of associating traces with wrong plaintexts.

Algorithm 6.1: Applet test 1 (S-box one iteration)

Input: Plaintext P, Cipher key k (stored in the smart card)
Output: Intermediate value V

```
Read plaintext P from card reader
Compute Value=SubByte(P xor k)
V=0
V=Value
Send V to card reader
```

6.4.2 Processing Times

For the JCOP30, the required time to run the applet test 1 is equal to 1,247.64 µs. This time was measured from the point the card reader ends the command transmission to the time the card reader receives the start of the response from the smart card. Using a sampling rate of 500 MS/s, it would be possible to acquire up to 51 traces from the emissions associated with applet test 1. The inconvenience of this approach is that the POI does not suggest a stable point of attack and most of the time required by the applet corresponds to reading the plaintext from the Application Protocol Data Unit (APDU) buffer and to prepare the buffer for the response.

To increase the number of events where the card is using the result of the S-box, a Java applet test based on Algorithm 6.2 was implemented. Here the variable *V* is cleared and later the result of the SubByte transformation is written to it. This write command is repeated 20 times before the result is transmitted to the card reader. The value of 20 repetitions was selected discretionally to guarantee the measurements are recorded in Steps 3 and 4 of Algorithm 6.2 and not in Step 6, when the card is sending the response to the card reader.

Algorithm 6.2: Applet test 2 (S-box 21 iterations)

```
Input: Plaintext P, Cipher key k (stored in the smart card)
Output: Intermediate value V
    1: Receive plaintext P from card reader
    2: Compute Value=SubByte(P xor k)
    3: V=0
    4: V=Value
    5: Repeat 20 times Steps 3 and 4
    6: Send V to card reader
```

For the case of the applet test 2, the processing time is 4,929.64 µs. Combining this with the results for applet test 1, this means that each cycle of calculating the result of the S-box and writing into a variable takes approximately 181 µs. To identify one possible time of attack, the window time between 1,400 µs and 1,600 µs was selected. In this interval, at least, one complete cycle of clearing and writing into the variable the result of the SubByte transformation occurs. One important point to consider is that the programmer of the Java Cards has no control over garbage collection (GC) events and the Java virtual Machine (JVM). The GC could be invoked at any time and thus potentially displace the times during which each instruction is executed. In general, the results presented in Chapter 7 correspond to acquisitions running applet test 2 unless otherwise specified.

The JCOP41 Java Cards process information approximately two times faster than the JCOP30 Java Cards. On average it takes 1.43 ms to run applet test 1 and start the reply while it takes 3.318 ms to run applet test 2 and respond. The difference between the two times (1.888 ms) corresponds to the time required to run 20 cycles of clearing and writing the variable *V*. This means each cycle requires 94.4 µs which is equivalent to 1,280 clock periods from the card reader.

The end of the command sent from the card reader to the smart card with the plaintext that the card processes looks the same for the JCOP30 and the JCOP41. Figure 6.4 illustrates the shape of different plaintexts going from 0x00 on the bottom to 0x0F on the top. The ISO/IEC 14443-3 standard (ISO/IEC, 1999b) indicates a 16-bit cyclic redundancy check (CRC) is appended to the end of the communication. This attachment makes the end of the command shown in the Figure 6.4 appear different from a consecutive binary pattern. According to this standard, the end of the command occurs 256 clock cycles (18.89 μs) after the transmission of the last bit. This means that the red line can be considered the approximated time where the card starts processing the information. The start of the response from the JCOP41 is shown in Figure 6.5, where an early start of the response for the cases when the card processed the plaintexts 5, 14 and 15 is visible.

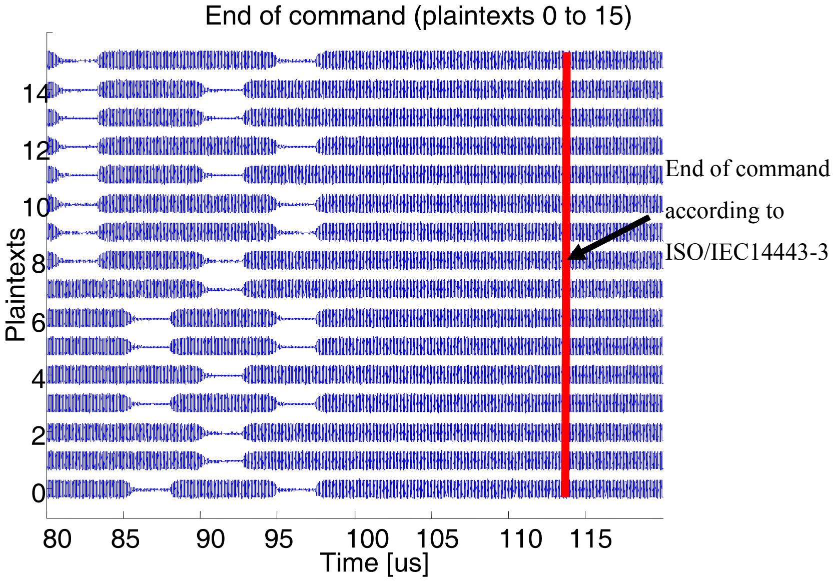


Figure 6.4: End of command from the card reader to the smart card JCOP41 for different plaintexts

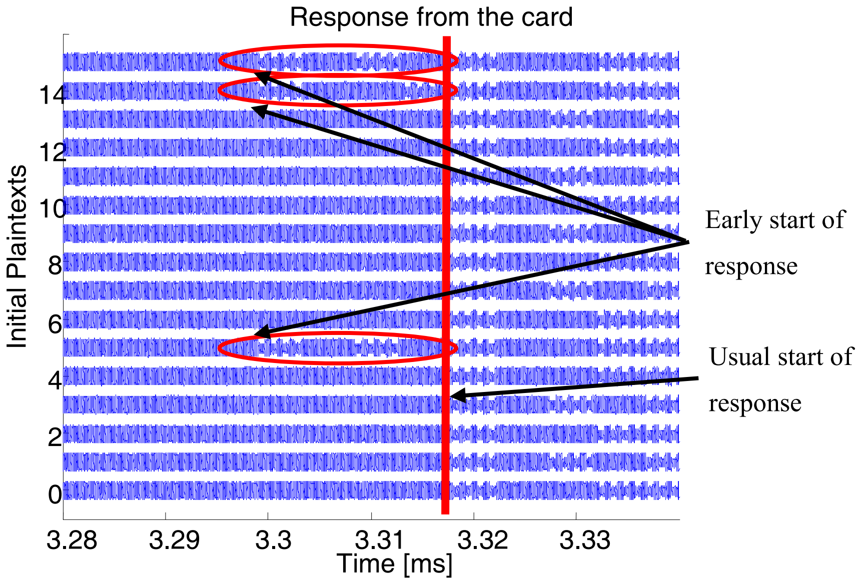


Figure 6.5: Start of response from the smart card JCOP41 to the card reader

In Table 6.1 the processing times for the JCOP30 and JCOP41 are compared when they run the applet test 1 and the applet test 2. For JCOP30, the responses of applet test 2 occur around 4,929.64 μs. In the case of JCOP41, the start of the responses takes place at 3.299 ms or 3.318 ms. This time depends on the instant the card finishes computing the results and is ready to start transmitting the response. Figure 6.5 exemplifies this situation. In the figure, the responses associated with the plaintexts 5, 14 and 15 occur earlier than the other plaintexts; however, this behaviour is almost arbitrary and we could not determine any pattern after analysing 50,000 traces.

Table 6.1: Processing time for the small testing applet

Card	Program	Time of end command [μs]	Time of start response [μs]	Processing time [μs]
JCOP30	Applet test 1	97.36	1,345	1,247.64
JCOP30	Applet test 2	97.36	5,027	4,929.64
JCOP41	Applet test 1	97.36	1,430	1,332.64
JCOP41	Applet test 2	97.36	3,299	3,201.64

6.5 Trace Acquisition

The fast frame function from the oscilloscope supports the capture of multiple traces. Each acquisition records a specific number of traces over a given time period. The limit for the oscilloscope used is 32 M samples (the maximum storage); for example, it is possible to acquire 1 trace of 32 M samples or alternatively up to 64,000 traces of 500 samples.

To capture the signals from the smart cards studied, different sampling rates were used. In Chapter 7 sampling rates of 250 and 500 MS/s are reported. For Section 7.1 a sampling rate of 250 MS/s, the GMR probe described in Chapter 5, and the JCOP30 running applet test 2 were used. Here five sets of 40 μs were obtained to cover the interval from 1,400 μs to 1,600 μs. Each set contained 39,936 traces captured on 13 acquisitions of 3,072 traces each. For the test results presented Section 7.2 a sampling rate of 500 MS/s using the GMR probe on the JCOP30 running the applet test 2 were used. There were 5,888 traces captured per acquisition. To capture the interval from 1,400 μs to 1,600 μs it was necessary to acquire 20 sets of 10 μs. For the test results presented Section 7.3 a sampling rate of 500 MS/s was used as well. The GMR probe and the JCOP41 card running the applet test 2 was analyzed. Each acquired trace was divided into smaller files of 1.18 μs. Each reduced file contains the data related to 16 clock cycles from the card reader. The rationale to have segments of 16 clock cycles was influenced by preliminary results for correlation analysis in the frequency domain where a different number of clock cycles was used to transform the EM traces from the time domain to the frequency domain, namely 8 cycles, the length for better preliminary results.

6.5.1 The Faraday Cage

One of the customized modifications made to better understand the possible side channel of the smart cards was removing the IC chip from the card and connecting it to the card containing the antenna using two wires. The main reason for this was to move the smart card microcontroller and the EM probe away from the card reader and its strong EM field. Having the IC chip away from the card reader in fact reduces the effects of the card reader; however, other signals also affect the measurements and the IC chip was placed inside a grounded metallic box emulating a Faraday cage. In Figure 6.6, a segment of one of the setups used in this research is presented where the IC chip was removed from the smart card and placed inside this "Faraday cage". Inside the box and on top of the IC chip the GMR probe can be seen the Altera DE2 board used to implement the triggering circuit is located next to the coordinates table. In this experiment, the GMR probe was manually placed on top of the IC chip inside of our Faraday cage and 9 different positions for each direction in the axis of sensitivity (X and Y) were evaluated. The area corresponding to the chip was divided into 9 sections, left, center, and right with respect to the X-axis and top, medium, and bottom with respect to the Y-axis. Then the middle point between pins 2 and 8 and 1 and 7 was the reference point on the probe and was positioned in each of the 9 sections. The position that seems to return the best results was with the probe in the top center position.

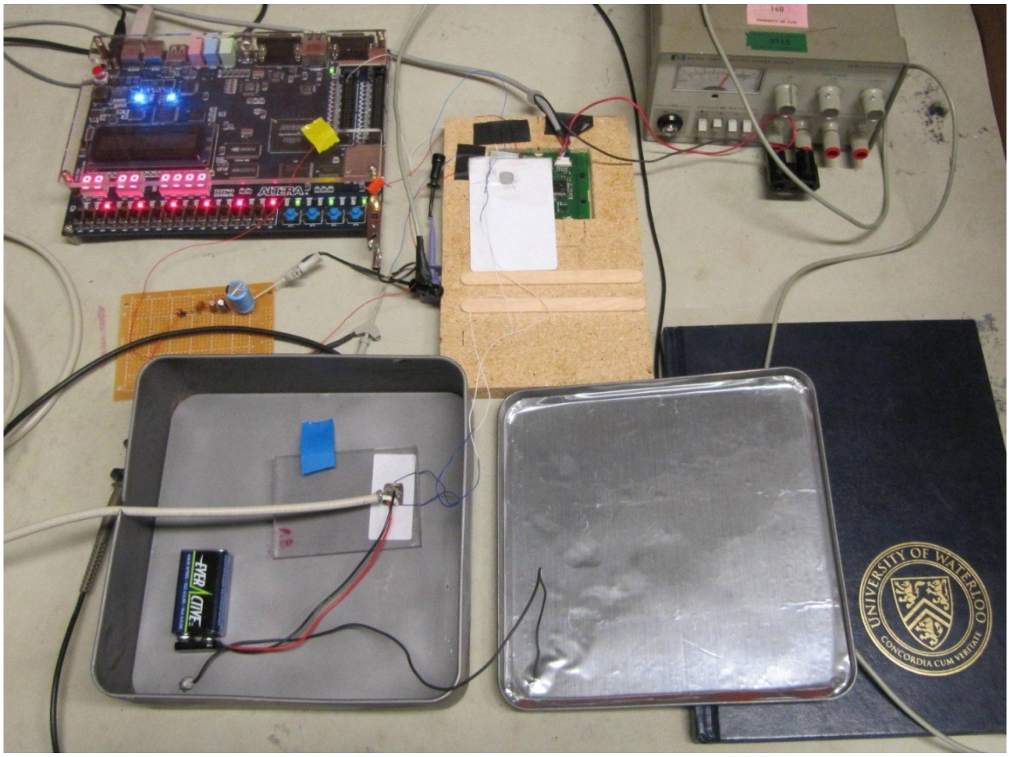


Figure 6.6: Segment of the setup used for analyzing the smart cards.

6.6 Comparison to Previous Research and Summary

An experimental setup for studying the side channel from Java Smart cards is presented in this chapter. First, some modifications made to a commercial card reader to improve the quality of the acquired traces are described. An axis table that manually sets the EM probe into different positions above the smart card was built. This approach differs from other research where the placement of the probes is automatic like described in (Quisquater and Samyde, 2001) or completely manual, without a systematic scanning as indicated in this research (Mangard, 2003). Other characteristics of the setup consisted of replacing the power supply and clock generator from the card reader. Initially the card reader was connected to Vcc and GND from a USB port, but the connections were modified to bypass the voltage terminals from the USB port and connect a laboratory power supply. In the case of the clock generator, it was removed and replaced by a stable signal generator. Triggering the oscilloscope from a smartcard working in contactless mode is difficult because the cards do not have contact points or external signals that could be used to send information to the oscilloscope and the trigger depends on the communication channel between the card reader and the card. In (Mangard, 2003), the author could not find a trigger for the oscilloscope based on the far field radiated emissions and they used a trigger connected physically with the card reader due to the impossibility of locating a trigger for the oscilloscope based on the radiated emissions. In other analyses where the smart cards worked contactlessly this difficulty was solved by using custom and freely programmable RFID readers (Kasper et al. 2009), (Kasper et al. 2011), (Oswald and Paar, 2011). In this research a commercial card reader was selected and a decoder was developed that in parallel to the smart card receives all the commands and triggers the oscilloscope when a programmed sequence of commands occur. Although in the experimental test, the oscilloscope’s high resolution mode returned slightly higher correlation than sampling mode, sampling mode allows capturing double the number of traces compared to the high resolution mode. Another element that helped in the search of a possible side channel on the Java Cards is removing the IC from the card and placing it away from the card reader inside of a metallic box that helps to reduce the effects from the card reader. Research such as (Carluccio et al. 2005) attempted an attack on a DESFire by removing the IC chip from the card, but given the unsuccessful results using DEMA and the inconvenience of having to examine the reader's antenna signal at every single measurement to obtain the plaintext used, they chose to build a custom RFID-reader. With respect to using a Faraday cage, some researchers have incorporated this element into their setups. In (Quisquater and Samyde, 2001) a customised socket was used to place a contact based smartcard inside of a Faraday cage while the card reader was outside. Other research such as (Plos et al. 2008) placed a smart card based on the microcontroller ATmega163 and the card reader inside a metallic box and they were able to reduce the number of traces required for a successful attack from 70,000 traces to 17,500 traces.

The sampling rate used for acquiring power or EM traces can affect the side channel analysis outcomes. In Chapter 5, it was illustrated that when using higher sampling frequencies, the correlation values increased. In previous research different sampling rates have been successfully used in the study of side channel analysis of the smart cards. For example, resolutions of 250 MS/s were used to analyse the power samples from an 8-bit microcontroller (Mangard et al. 2007) and 500 MS/s were used in other research (Moradi et al. 2009) (Oswald and Paar, 2011). Other works had opted for higher sampling rates like in (Carluccio et al. 2005) (Lu et al. 2009b) where the sampling rate of 1 GS/s was used to analyse emissions from a smart card and ASICs respectively. However, with an oscilloscope's constrained memory there is a trade-off among sampling rate, length of acquisition and the number of traces acquired. In this research, resolutions of 250 MS/s and 500 MS/s were considered for the study of the smart cards. The next chapter describes the empirical results of using the setup developed in this chapter.

Chapter 7

Empirical Analysis of the Java Smart Cards

A synopsis of the results obtained after analysing the JCOP30 and JCOP41 Java Cards is presented in this chapter. It is divided into 3 main sections and a summary. Section 7.1 presents the results obtained after analysing the EM power traces obtained from the JCOP30 smart card when the IC chip was removed from the card; Section 7.2 expounds the results obtained after analysing the EM traces captured from the JCOP30 smart card without any modifications made to the card itself; and Section 7.3 analyzes the results from analysis of the EM traces obtained from the unmodified JCOP41 smart card. In the three cases, the analyses considered the EM traces in the interval from 1,400 µs to 1,600 µs after the trigger. Although many experiments and analyses using different clock cycles (6, 8, 12, 16, 24, 30, 32, and 64) were investigated, due to limited space, the results presented in this chapter focus on intervals of 8 clock cycles (0.589 µs) at the point where the analyses returned the best results. Considering the difficulties of acquiring EM traces from the Java Cards that successfully retrieve the correct key using a commercial inductive EM probe, a number of alternative probes were tested. Those experiments included inductors of different sizes, wire gauges, number of loops, VCR heads for VHS tapes, and a probe that uses giant magnetoresistance sensors (Mateos and Gebotys, 2011). All the analyses presented in this chapter come from traces acquired using a GMR probe. For Section 7.1 the results were verified for keys 0x5C and 0x9E. For Section 7.2 the keys 0x03, 0x3D, 0x5C, 0x9E, A2, and 0xE3 were tested. Finally, for Section 7.3 the keys 0x03, 0x9E, and 0xA2 were verified.

7.1 Java Card JCOP30 with Modifications

In an early stage of this research, EM traces acquired from a JCOP30 Java Card without modifications were analysed and those analyses did not succeed in returning the correct keys. Different EM probes were used and the card was placed in different positions but the results consistently failed in guessing the correct keys. These difficulties motivated a modification to the smart card that consisted in removing the IC chip from the card and placing it away from the card reader and its strong magnetic field. With this change and using the GMR probe, it was possible to retrieve the correct key used. This section describes the results obtained after analyzing the JCOP30 with the IC chip removed from the card and placed inside a Faraday cage as explained in Chapter 6, The results presented in this section were obtained by analyzing 39,936 traces (13 acquisitions of 3,072 traces each) acquired with a sampling rate of 250 MS/s using the GMR probe without a preamplifier, when 39,936 plaintexts were processed using the hexadecimal value 0x5C as the small part of the key, using the JCOP30 running applet test 2. With the selection of a lower resolution, 250 MS/s instead of 500 MS/s, it was possible to cover the double of time with the same acquisitions. The window of time captured was from 1,400 μs to 1,600 μs.

7.1.1 Time Domain Analysis

As explained in Chapter 3, correlation analysis measures the level of association between the EM traces when the studied circuit is processing some cryptographic information and the hypothetical power consumption model. The result of this analysis is a matrix where the columns correspond to the sampling times and each row corresponds to a possible key guess, row 0 corresponds to key guess 0, row 1 to key guess 1, etc. The relevant information for this experiment occurs in the interval referred to as 1,498.5 μs to 1,499.1 μs that corresponds to 8 cycles of the card reader. The sampling rate of 250 MS/s corresponds to 146 samples (columns of the correlation matrix). This time region was found after several acquisitions and analyses across the interval 1,400 μs and 1,600 μs searching for a region were the correlation analysis returned the correct key. Results were verified using different keys such as 0x03, 0x3D, 0x5C 0x9E, and 0xE3.

For the traces studied here, the result from the correlation analysis in the time domain is a matrix where row 92 (corresponding to the correct key guess 0x5C) has the highest value in the matrix and some of its elements are the largest in their columns. This means that the key guess ranked in first place corresponds to the key used to process the data. A 3D plot from the correlation matrix where the values for the correct key guess are marked in black while all other key guesses are represented in a different color from blue to red is presented in Figure 7.1. Key guess 92 has a higher correlation than its neighbours for all time samples, but this is not easy to observe in Fig. 7.1.

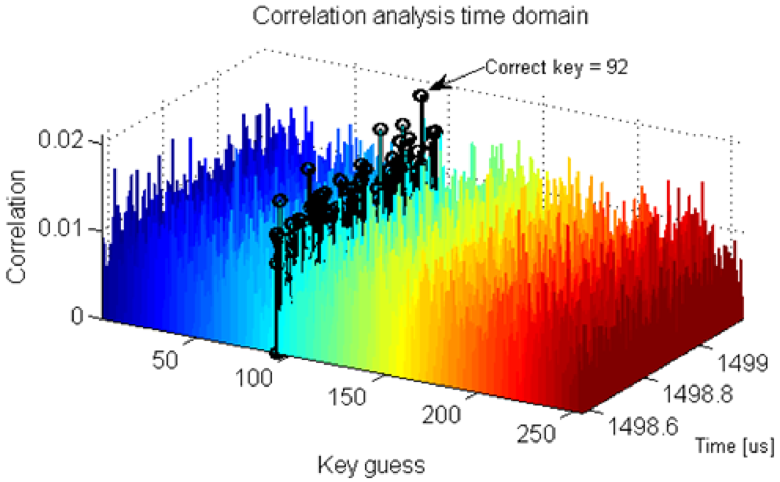


Figure 7.1: Representation of the correlation analysis matrix in the time domain using 39,936 traces.

A projection of the correlation matrix showing the time interval from 1,498.5 µs to 1,499.1 µs is presented in Figure 7.2. In this plot, each color matches a key guess which is superimposed on each time sample on this plot. The maximum correlation for each time sample belongs to the key guess 92. The magnitude of the maximum correlation coefficient is 0.01959 and occurs at 1,498.65 µs. The figure illustrates that the correlation of the correct key guess (black circle with star inside it) is bigger than the other higher correlation keys at different times. The difference between the maximums of the key guesses ranked in first and second place (key guess=26 at 1,498.59 µs) is 2.6%.

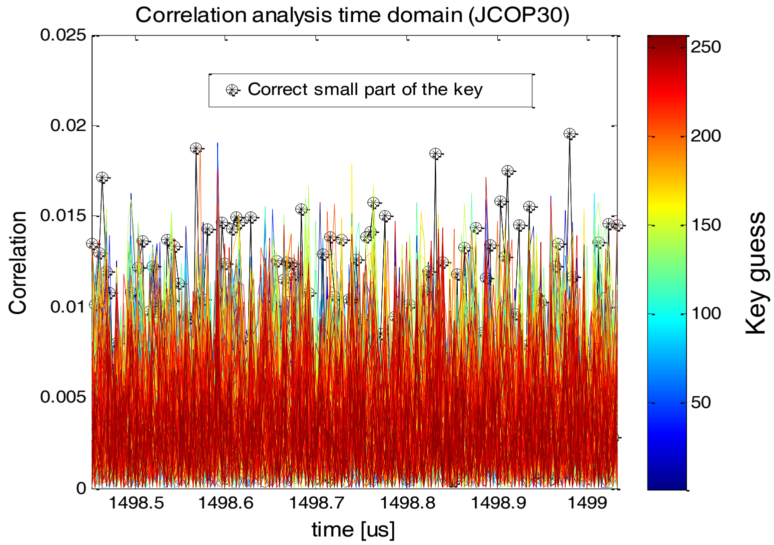


Figure 7.2: Projection from the correlation analysis matrix in the time domain using 39,936 traces.

The projection of the key guess vs. correlation showing the maximum value of the correlation matrix for each key guess over the entire time range is illustrated in Figure 7.3. The maximum correlation for the correct key guess is 2.98 bigger than the mean of all key guesses maximums.

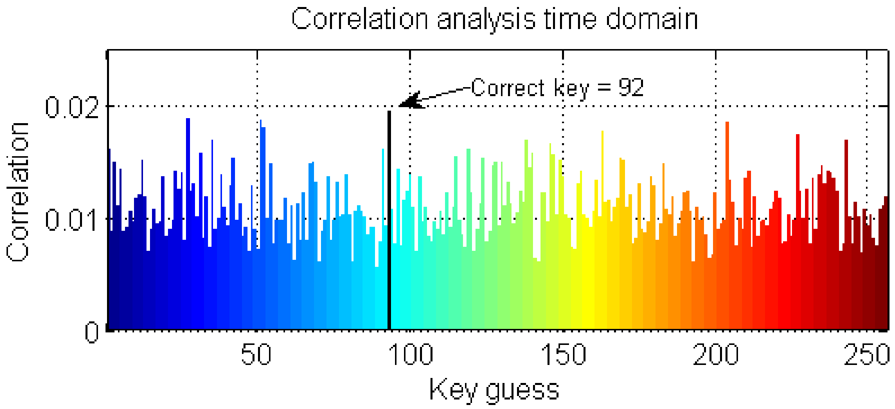


Figure 7.3: Projection of the correlation analysis matrix in the time domain using 39,936 traces showing the maximum correlations for each small part of the key guess in the interval 1,498.5 μs to 1,499.1 μs.

Using SPA, the average of all traces according to their Hamming weight is presented in Figure 7.4. There are 8 maximum amplitude peaks around 5 mV which are visible. These values are related to the clock cycle of the card reader. The figure illustrates zooming into one part of the figure around 1,498.65 µs, where the correct key guess reaches its maximum correlation. The sequence of the Hamming weights measured at this point is 0, 1, 6, 3, 5, 4, 2, 7, and 8. The values 6, 5, and 2 are misplaced. According to the values shown in the figure, the difference in the voltage measured between the traces corresponding to the Hamming weights HW=7 and HW=8, is 175 µV while the difference measured for HW=0 and HW=1 is approximately 10 µV. Table 7.1 shows the distribution of the Hamming weights according to the number of traces used. The next section examines the frequency domain analysis.

Table 7.1: Distribution of the 39,936 traces used according to their Hamming weight

HW	# of traces	Percentage
0	156	0.39
1	1,248	3.13
2	4,368	10.94
3	8,736	21.88
4	10,920	27.34
5	8,736	21.88
6	4,368	10.94
7	1,248	3.13
8	156	0.39

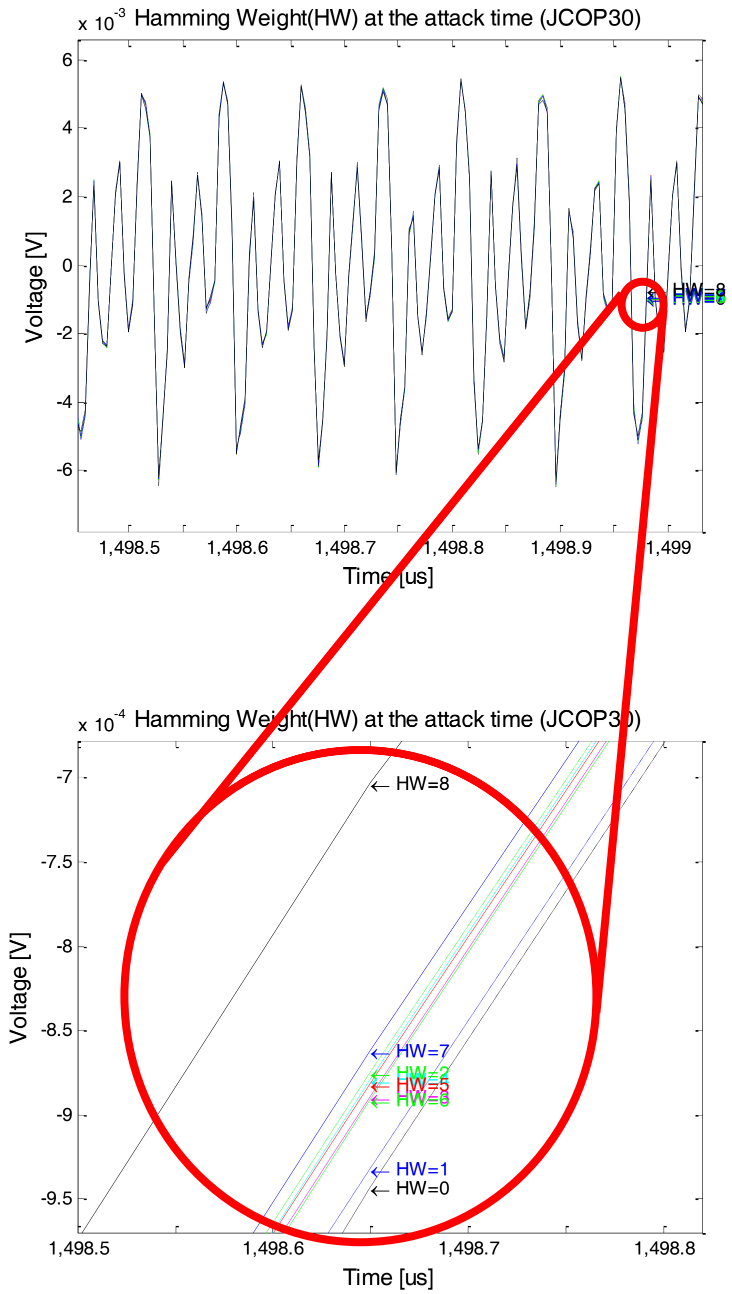


Figure 7.4: Hamming weight of the acquisition traces at the maximum correlation. On top the interval 1,498.5 µs to 1,499.1 µs and in the bottom a zoom to the interval 1,489.5 µs to 1,498.8 µs.

7.1.2 Frequency Domain Analysis

Correlation analysis in the frequency domain measures the grade of similarity between the hypothetical power consumption model and the power spectrum from the measured signals. The result is a correlation matrix where the rows are related to the hypothetical key guesses and the columns to each one of the frequencies from the power spectrum signal. The traces analyzed in this section are the same traces that were studied in Section 7.1.1. These traces were transformed into the frequency domain using the FFT. The sampling rate of the setup used was 250 MS/s and the range of frequencies went from DC to 125 MHz. Refer to Chapter 2 for more details on the procedure to obtain the correlation matrix.

The maximum value in the correlation matrix obtained from the analysis in the frequency domain is in row 92 corresponding to the correct key guess 0x5C and the column associated with the frequency of 112.85 MHz. In Figure 7.5, the correlation matrix is represented, with one axis representing the 256 possible key guesses and in the other axis the frequencies from DC to 125 MHz. The correlation values for the correct key are highlighted in black while the values for each other key are in colors from blue to red.

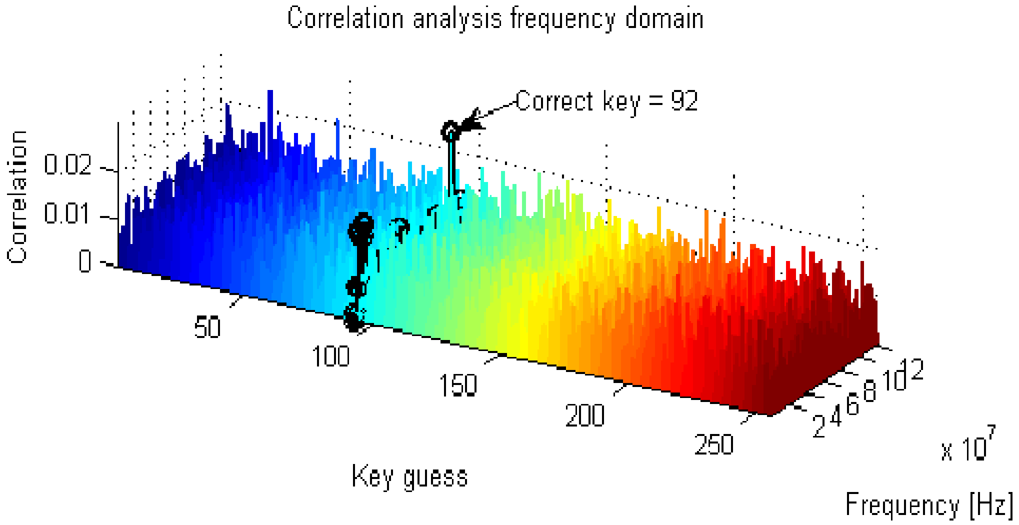


Figure 7.5: Representation of the correlation analysis matrix in the frequency domain using 39,936 traces.

The superimposed correlation matrix showing the frequency and correlation axis is illustrated in Figure 7.6. The color of traces corresponds to each key guess used while the black trace with a star shape corresponds to the correct small part of the key used. In this case, the maximum correlation for the correct key guess occurs at 112.85 MHz. It is also clearly observed that at 12.15 MHz, the correlation for the correct key is higher than the correlation of the other keys at that frequency.

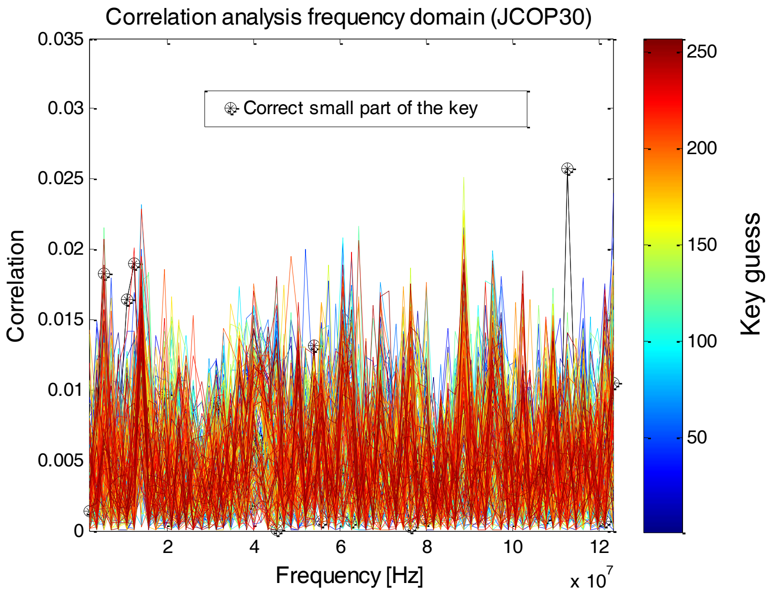


Figure 7.6: Projection from the correlation analysis matrix in the frequency domain using 39,936 traces.

The projection of the key guess vs. correlation is shown in Figure 7.7. The correlation for the correct key (r=0.02565) is higher than any other key guess. The second maximum correlation corresponds to the key guess 146 and it is 3.15% smaller than the first. In the case of the maximum correlation, it is 3.2924 standard deviations bigger than the mean of all other maximums.

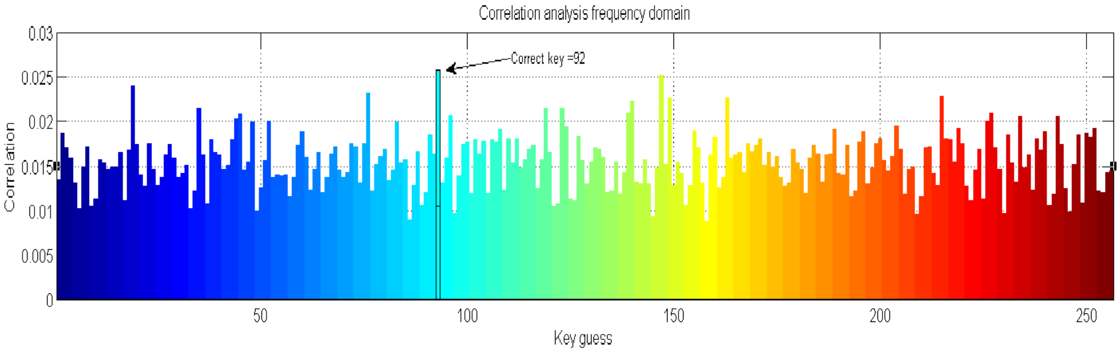


Figure 7.7: Projection of the correlation analysis matrix in the frequency domain using 39,936 traces showing the maximum correlations for each small part of the key guess in the interval 1,498.5 µs to 1,499.1 µs.

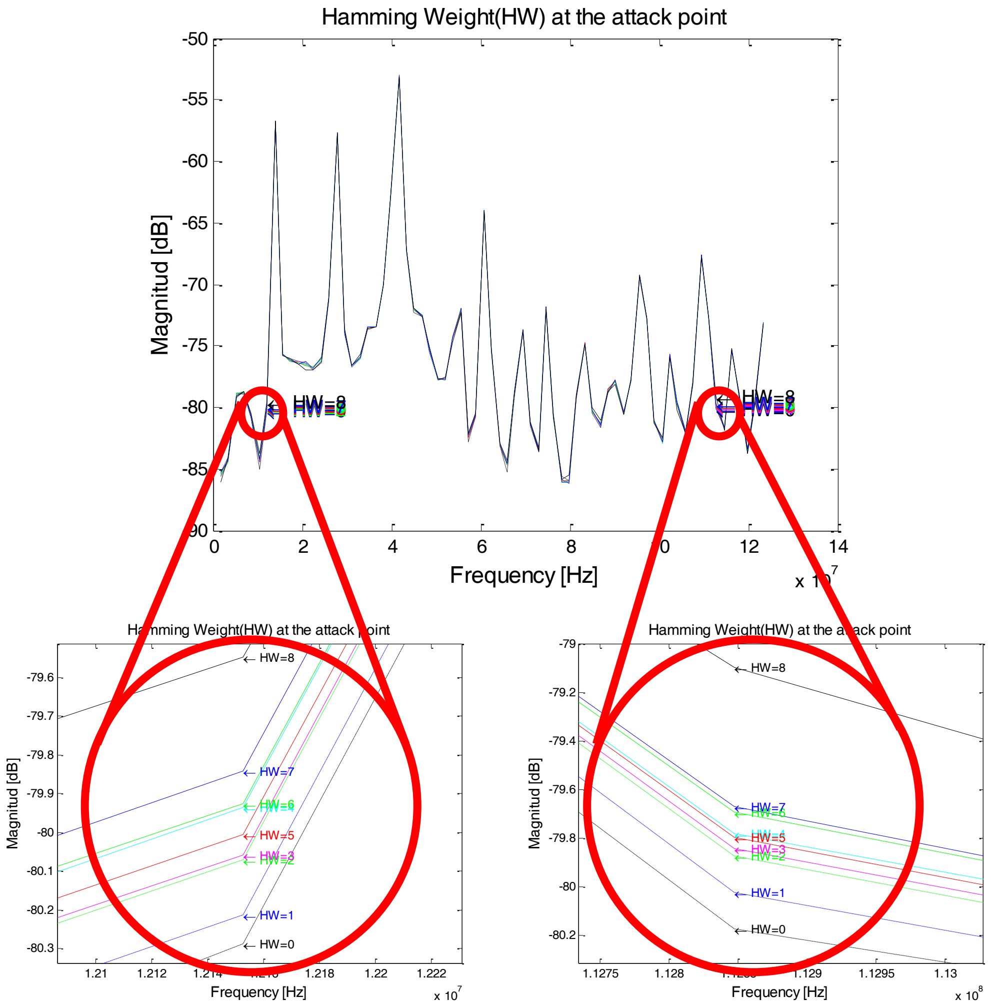


Figure 7.8: Hamming weight analysis in the frequency domain.

Using SPA, the means of all FFTs from the traces for each Hamming weight are presented in the Figure 7.8. In the upper side of the figure some picks are visible at 13.56 MHz, 27.12 MHz, and the highest of all at 40.68 MHz. The bottom part of the figure zooms in on two frequencies 12.15 MHz and 112.85 MHz. At both frequencies, it is observable that the EM power measured from the circuit has a slight resemblance to the expected power consumption where the average of each Hamming weight has a sequentially increasing/decreasing value. In the measured signals, the sequence is 0, 1, 2, 3, 5, 4, 6, 7, and 8. The misplacement of HW=4 and the use of logarithmic scale contributes to a correlation of around 0.025 for 112.85 MHz and 0.016 at 12.15 MHz.

Another important detail in this graph is the magnitude of the power measured. Considering that -80 dB corresponds to 10 nW and -80.2 dB to 9.54 nW, it is possible to infer from the graphs that the difference in measured power consumption when no bits change (HW=0) and when one bit changes (HW=1) is approximately 450 pW at 112.8 MHz.

7.2 Java Card JCOP30 without Modifications

This section presents the results obtained after analyzing the JCOP30 smart card without modifications in the time and frequency domain. For this analysis 5,888 traces were captured in a single acquisition with a sample rate of 500 MS/s using a GMR probe and a wideband amplifier when the smart card runs the applet test 2. These traces were captured later than those in the previous section, and due to time limits imposed on this research, a larger set of traces was not possible. In this setup, no Faraday cage was used. The results presented correspond to the key=0xA2 (162d) and were verified for other keys (0x03, 0x3D, 0x5C, 0x9E, and 0xE3) with similar results. The traces acquired and analysed for this experimental setup included 20 sets of 5,888 traces over 10 µs intervals that cover from 1,400 µs to 1,600 µs. The relevant information from the analysis occurs in the time between 1,498.5 µs and 1,499.1 µs corresponding to 8 cycles of the card reader clock. The axis table described in Chapter 6 was used to position the GMR probe. In Chapter 5, the reference point to position the GMR probe was discussed a the point between the pins 1, 2, 7, and 8 from its MSOP8 package. This point was positioned on the top center position corresponding to the area of the IC as shown before in Figure 6.3.

7.2.1 Time Domain Analysis

The result of correlation analysis in the time domain for this setup is a correlation matrix with 256 rows (associated to the key guesses) and 294 columns associated with the 1,498.5 µs to 1,499.1 µs interval. The values of this correlation matrix are illustrated in Figure 7.9. It indicates that the maximum correlation among the key guesses corresponds with the correct key used. In the figure, each color trace corresponds to the correlation values of a particular key guess and the correct key=162 (0xA2) is colored black with a star on it.

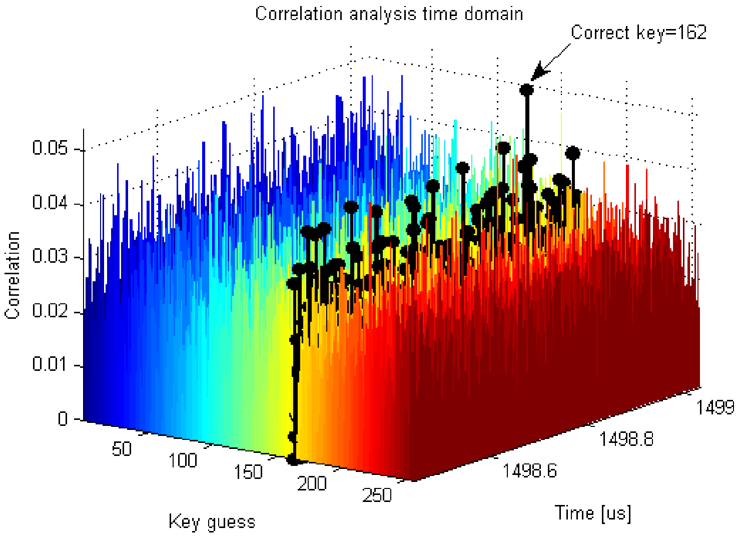


Figure 7.9: 3-D representation of the correlation analysis matrix in the time domain for the JCOP30 smartcard without modifications.

The correlation matrix has a number of values close to 0.05 and the 3D perspective affects the perception of the results. To compare these values, Figure 7.10 provides a projection of the correlation values across the time axis. In the plot, each color corresponds to one of the key guesses. The correlation values corresponding to the correct small part of the key (0xA2=162) are highlighted using a black trace with a star. At some points it is visible that the correlation values for the correct key are bigger than other key guesses. The maximum correlation is 0.05407 and it occurs at 1,498.92 µs. The difference between the first and second place (key guess=26 at 1,498.74 µs) is 2.5%.

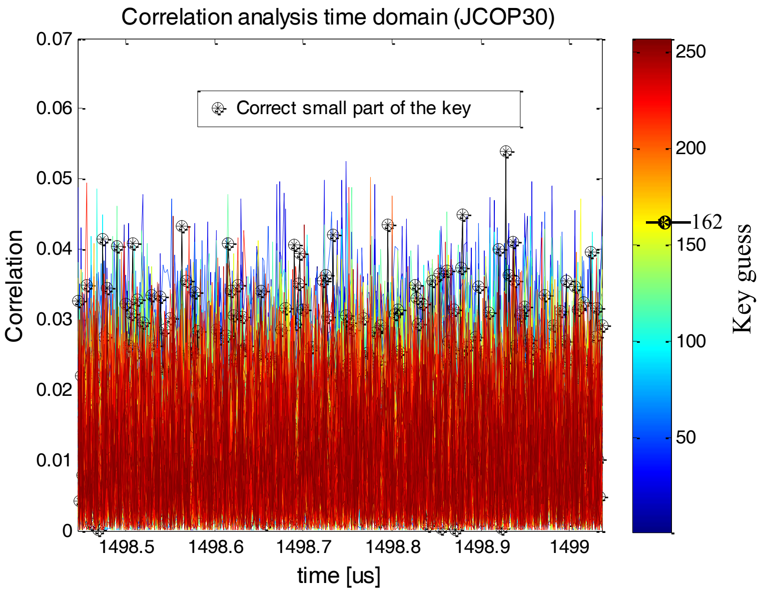


Figure 7.10: Projection of the correlation matrix for the time domain analysis using 5,888 traces showing the correlation values vs. time.

A graph containing the maximum correlations for each of the key guesses over the entire time interval is shown in Figure 7.11. In this case, the maximum value for the correct key is 3.27 standard deviations bigger than the mean of all key guess maximums.

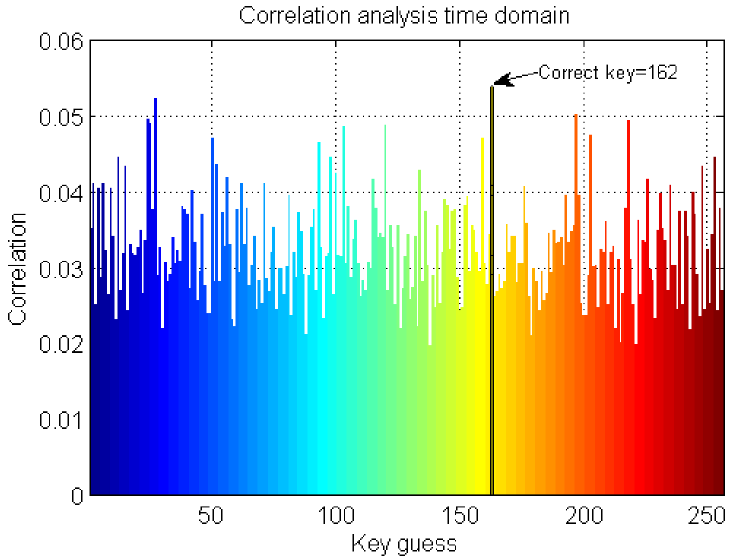


Figure 7.11: Projection of the correlation matrix for the time domain analysis using 5,888 traces.

In the case of the other keys tested, the correlation magnitudes were similar to the ones obtained for key 0xA2 and they are presented in Table 7.2. Although the correct key was not guessed in all cases, it was returned among the first 12 ranks.

Table 7.2: Test results for other keys used JCOP30

Key used	Time [μs]	Rank	Max correlation correct key
0x03	1,492.48	2	0.0544
0x3D	1,496.02	3	0.0502
0x9E	1,497.88	1	0.0529
0xA2	1,498.92	1	0.0540
0x5C	1,493.73	5	0.0534
0xE3	1,494.79	12	0.0473

The average of the 5,888 traces according to their Hamming weight using the power model for the correct key guess and a sampling rate of 500 MS/s is plotted in Figure 7.12. In the graph, the 8 clock cycles with the highest voltage in each period above 50 mV are distinguishable. The bottom part of the figure shows the time where the correlation analysis returned the maximum correlation (0.05407 at 1,498.928 µs). At this point the order of the Hamming weights from bottom to top is 8, 0, 2, 1, 3, 6, 4, 5 and 7. Apparently, the average Hamming weights that are out of sequence are 8, 2 and 6, but the ones corresponding to 0, 1, 3, 4, 5 and 7 are in the correct order and those represent the 77.7% of the traces used. With this setup the difference calculated between HW=0 and HW=1 is 164 µV. Table 7.3 shows the distribution of the Hamming weights according to the number of traces used; here the plaintexts were repeated 23 times from 0x00 to 0xFF.

Table 7.3: Distribution of the 5,888 traces used according to their Hamming weight

HW	# of traces	Percentage
0	23	0.39
1	184	3.13
2	644	10.94
3	1,288	21.88
4	1,610	27.34
5	1,288	21.88
6	644	10.94
7	184	3.13
8	23	0.39

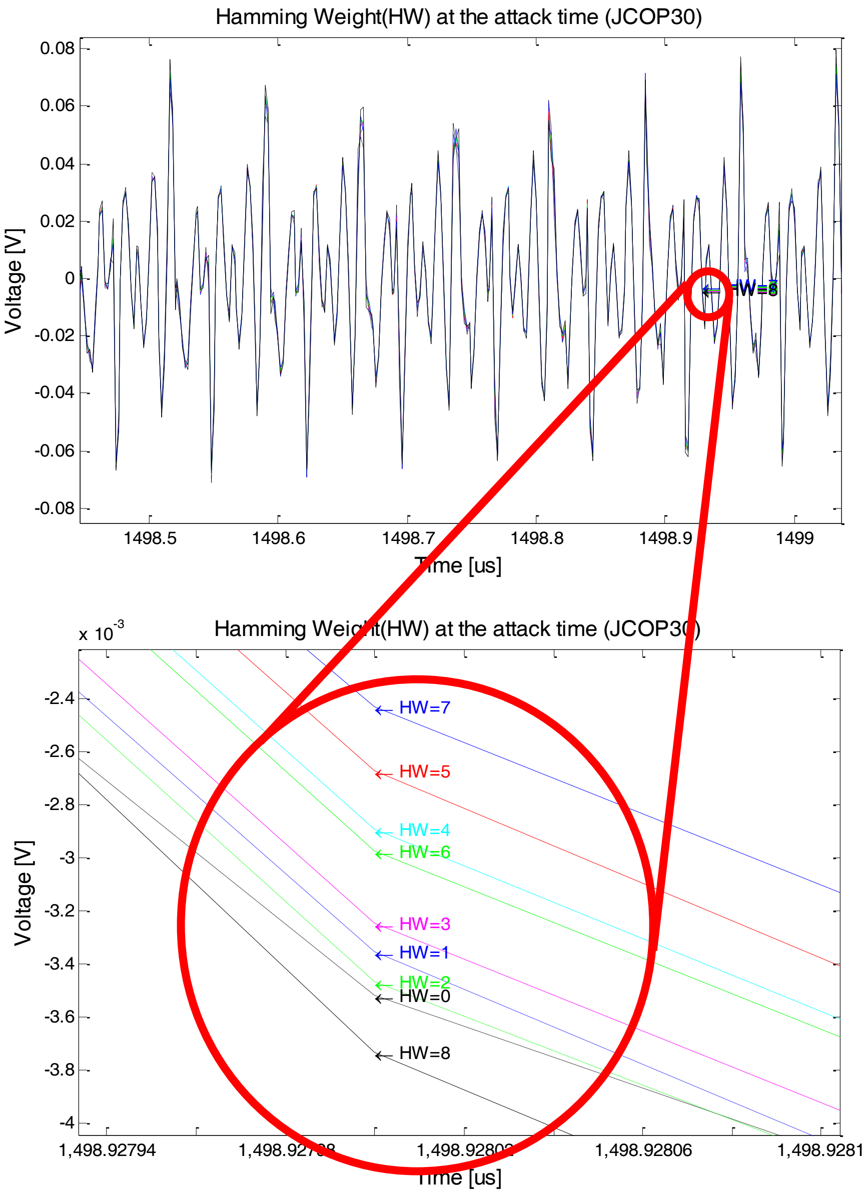


Figure 7.12: Average of 5,888 traces according to its Hamming weight using the power model for the correct key guess and a sampling rate of 500 MS/s.

7.2.2 Frequency Domain Analysis

The EM traces analysed in this section are the same signals studied in the previous Section 7.2.1 from times 1,498.5 μs to 1,499 μs but mapped into the frequency domain using the FFT. According to the sampling rate of 500 MS/s, the columns from the correlation matrix are associated with frequencies from DC to 250 MHz. After analysing the traces in the frequency domain, the resulting correlation matrix is represented in Figure 7.13. Its maximum correlation value (0.07) appears in row 162 corresponding to the key guess (0xA2) and is highlighted in black. This appears at a frequency of 170 MHz.

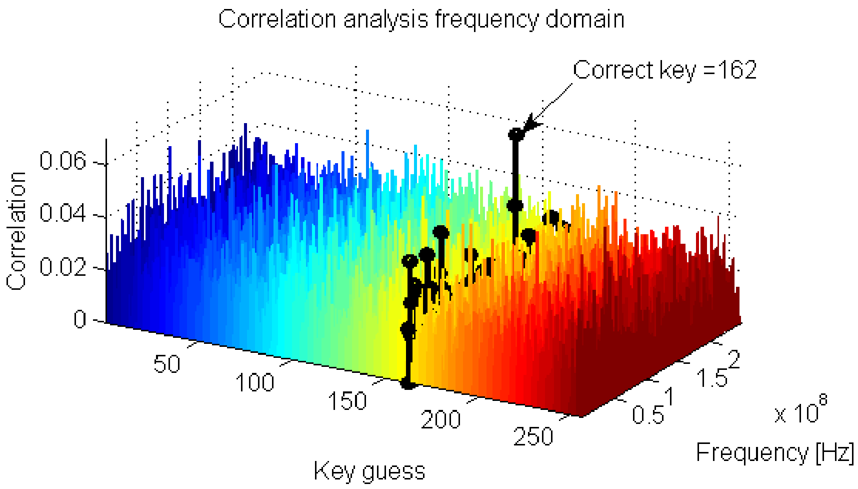


Figure 7.13: 3 D representation of the correlation matrix in the frequency domain using 5,888 EM traces.

A projection of the correlation matrix showing the frequency and correlation axis is presented in Figure 7.14. The maximum value from the matrix is 0.07016 which identifies the correct key and appears at a frequency corresponding with 170 MHz. The second maximum correlation corresponds with the key guess 249 (0xF9) at 54.4 MHz. In this case, the difference between first and second place is 9.79%, the magnitude of the maximum value. Analysing only the magnitudes from all key guesses at a frequency of 170 MHz, the difference between the first and the second largest values is 35.17%.

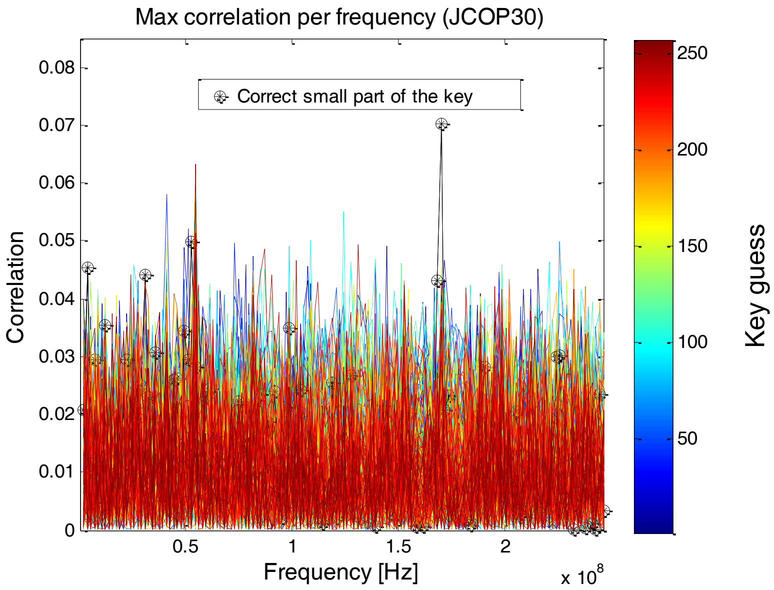


Figure 7.14: 2-D projection from the max correlation per frequency after using 5,888 EM traces.

The max correlations over all frequencies corresponding to each key guess are presented in Figure 7.15. For this set of values, the distance between the maximum value of the correct key and the average of all maximums is 3.73 standard deviations.

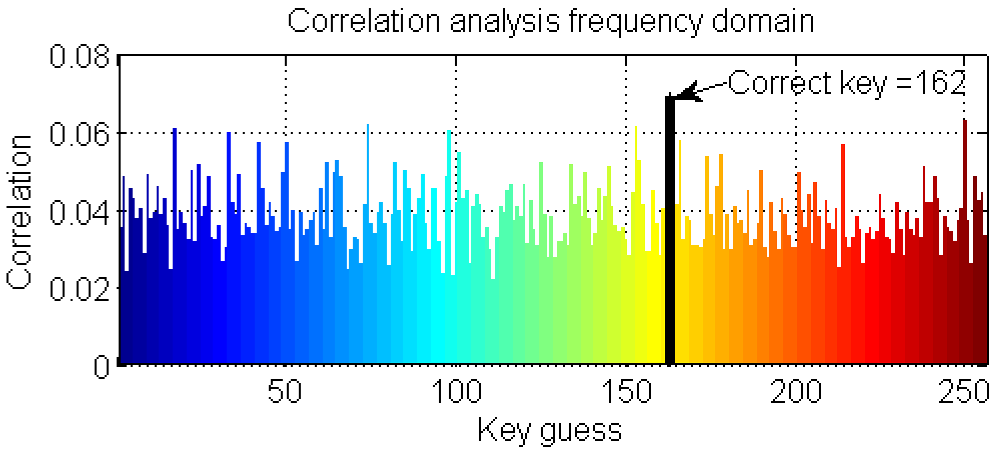


Figure 7.15: Projection of the Max correlations over all frequencies for the JCOP30 using 5,888 traces.

The analysis in the frequency domain for other keys showed smaller spikes than key 0xA2. Table 7.4 shows a synthesis of these results at the times where the correlation analysis, in the time domain returned the best results. Similar to the time domain analysis the correct key was not returned in all cases but in this case it was ranked among the first 5.

Table 7.4: Test results for other keys used JCOP30

Key used	Frequency [MHz]	Rank	Max correlation correct key
0x03	23	3	0.0610
0x3D	124	1	0.0644
0x9E	168	5	0.0592
0xA2	170	1	0.0701
0x5C	23	2	0.0603
0xE3	52	3	0.0615

The mean of all traces that share the same Hamming weight based on the hypothetical power model using the correct key guess are displayed in Figure 7.16. In the top part, the power spectrum of the signals is shown and a number of peaks on the multiples of the 13.56 MHz clock frequency (27.12 MHz, 40.68 MHz, 54.24 MHz, etc) are visible. In the bottom part, the area around the two frequencies, 170 MHz and 3.4 MHz where the correlation for the correct key was bigger than the other key guesses, is magnified. At 170 MHz the sequence of the Hamming weights is 1, 0, 8, 3, 2, 4, 7, 5, and 6; in this case 1, 8, 2, and 7 are misplaced. For the frequency of 3.4 MHz the Hamming weight sequence from bottom to top is 8, 0, 1, 3, 2, 4, 5, 6, and 7. The Hamming weights that seem to be out of order are 8 and 3.

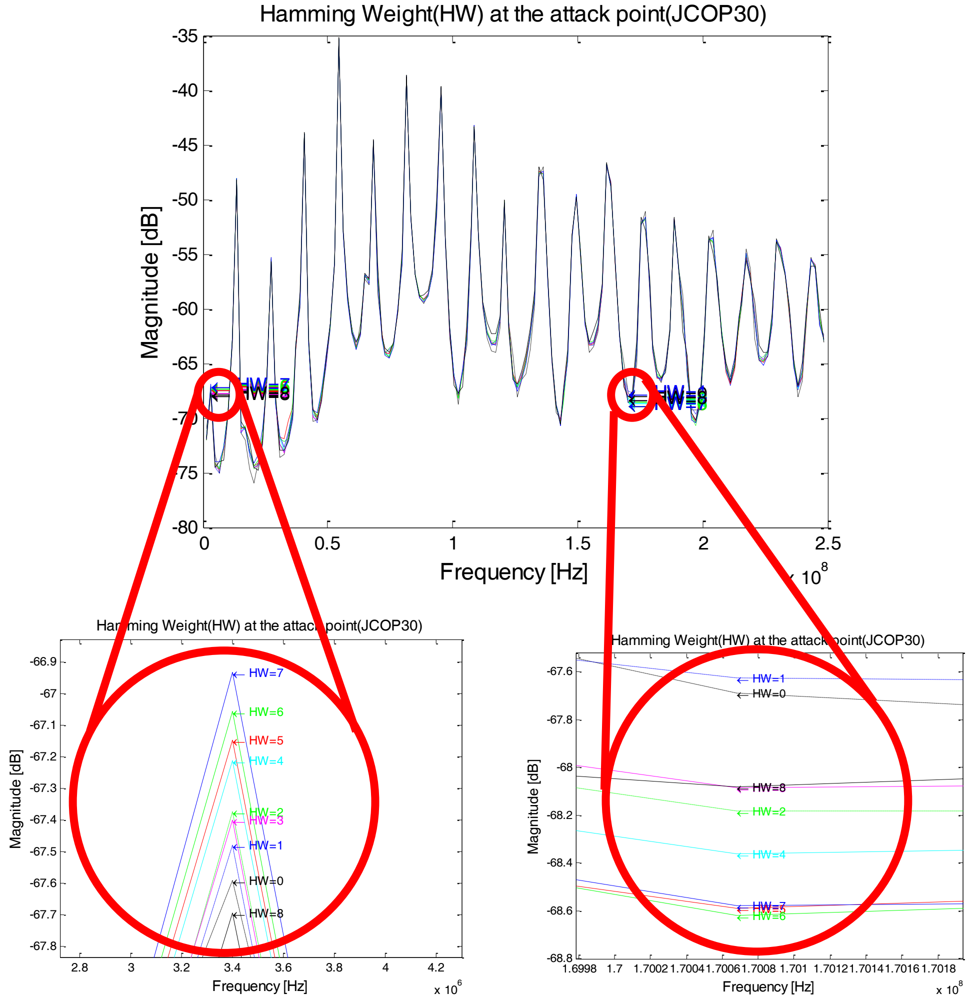


Figure 7.16: Power spectrum averages according to their Hamming weights.

7.2.3 Measuring the Effectiveness of the Analysis in the Time and Frequency Domain

In Chapter 3, two metrics were introduced to measure the effectiveness of an attack: accuracy and estimation. Accuracy uses the correct key information and the correlation matrix to quantify how close the key guess was to the correct guess. Defined in Equation (3-12), it is an exponential scale that uses powers of 2 to evaluate the results of a given analysis and helps to compare different attacks by quantifying the effectiveness of each attack. If the small part of the key is 1 byte long then when a key guess is ranked in first place, the accuracy is 1, if it is ranked in second place, the accuracy is 0.5, etc. In the case of *estimation,* it analyses the dispersion of the maximum correlations for the key guesses and when the magnitude of the maximum correlation is δ standard deviations bigger than the mean of all maximums the estimation will be 1 and in any other case, it is 0. This latter metric does not require any knowledge of the correct key.

Next, the results for correlation analysis in the time domain, frequency domain, and fast correlation analysis are compared using accuracy as a metric. In Figure 7.17 the results obtained by running the 3 respective analyses starting with one trace and adding traces until reaching 5,888 using a linear scale (left) are presented. The figure shows that using correlation analysis in the frequency domain it is possible to find the correct small part of the key guess with 386 traces and the result remains for up to 495 traces. Adding more traces to the analysis, the accuracy moves back to values lower than 0.001 to later come back to an accuracy of 1 using 3,596 traces. Reviewing the results for the frequency domain analysis, it was found that the frequency that returned the highest correlation from 386 traces to 495 traces was 25.5 MHz, while after 3,596 traces, it is 170 MHz.

For the case of correlation analysis in the time domain, the analysis reaches an accuracy of 1 with 4,155 traces and remains there before 4,180 traces. Then, there are some variations between 0.5 and 0.125 and after 5,000 traces it moves to one thousandth. With more than 5,526 traces, the accuracy moves to values between 0.25 and 1 (though most of the time it is at 1). Additionally, using Equation (3-11) and considering 5,888 traces, the noise floor is 0.0521. According to (Mangard, 2004), setting an $\alpha=0.9$ in Equation (3-10) provides a reasonable value for calculating the lower bound of the number of traces necessary for an attack. Using 5,888 traces, α=0.9 (z_α=1.282) (Mangard et al. 2007), and applying Equation (3-10), the result is a correlation of ρ=0.0236 and for α=0.995 (z_α=2.576), it is ρ=0.0475. The maximum correlation value r found experimentally (0.0547) is bigger than $\rho_{\alpha=0.9}$ and $\rho_{\alpha=0.995}$. This means that the attack in the time domain should be considered meaningful.

For the case of fast correlation in the frequency domain, only the component of 170 MHz was considered. In Figure 7.17 with the semi-log scale, the accuracy starts very small reaching its minimum 9.056×10^{-72} with 191 traces from there it starts rising with a few fluctuations and with 766 traces it reaches the accuracy of 1. It keeps fluctuating between 0.0039 and 1 and finally become stable at 1 after 2,437 traces. In the case where the correct key would be unknown, using estimation and a factor δ=4.6, 3,629 traces would be required to guess the correct key.

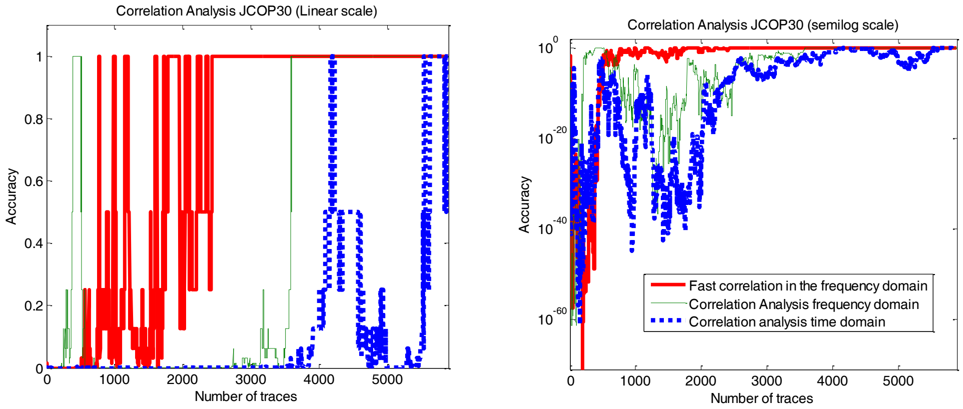


Figure 7.17: Comparison of 3 types of correlation analysis using accuracy as metric linear scale (left) and semi-log scale (right).

7.3 Java Card JCOP41 Without Modifications

In this part of the dissertation, the results obtained from the analysis of the JCOP41 when 5,888 EM power traces are acquired using the GMR probe are illustrated. The sampling rate used is 500 MS/s and the time reported corresponds to the 8 clock cycles between 1,535.8 μs and 1,536.4 μs. This interval returned the highest magnitudes for correlation analysis in the frequency domain. The hexadecimal value used as the small part of the key was 0x03. For this experiment the GMR probe and the wideband amplifier were used.

7.3.1 JCOP41 Time

Unlike the results of the previous two sections, it was not possible to find the correct key guess using correlation analysis in the time domain. A 3D representation of the correlation matrix is displayed in Figure 7.18. The key used for processing the data is highlighted in black with a circle on top. In the figure, there are other key guesses that have higher correlation values than the maximum for the correct key used.

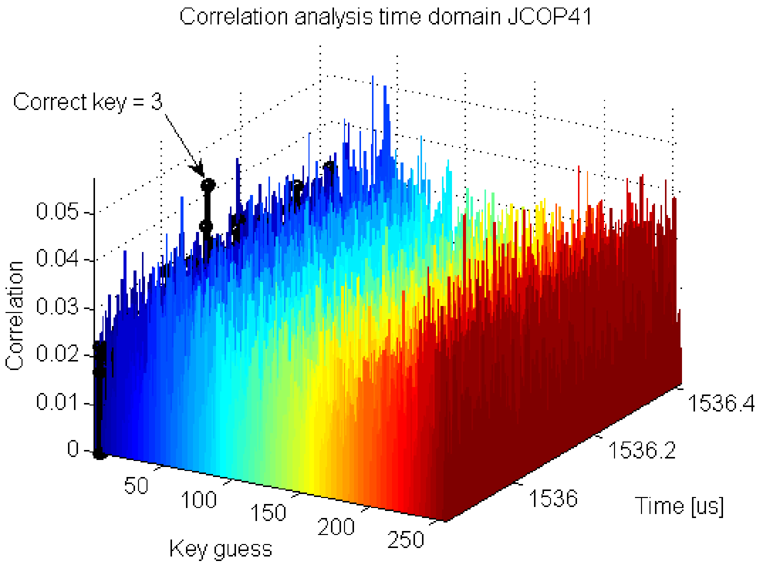


Figure 7.18: Representation of the correlation matrix in the time domain using 5,888 traces.

A projection of the correlation matrix from the time domain analysis is shown in Figure 7.19. The correct key is displayed in black with a star on it; however, there are a few points where the key guess overcomes the other values. At 1,536.1 μs with a correlation of 0.04266, the maximum value for the correct key guess appears. The maximum correlation in the interval displayed is 0.05726. It corresponds to a wrong key guess (234) and it is 4.23 standard deviations bigger than the average of the highest values for all key guesses.

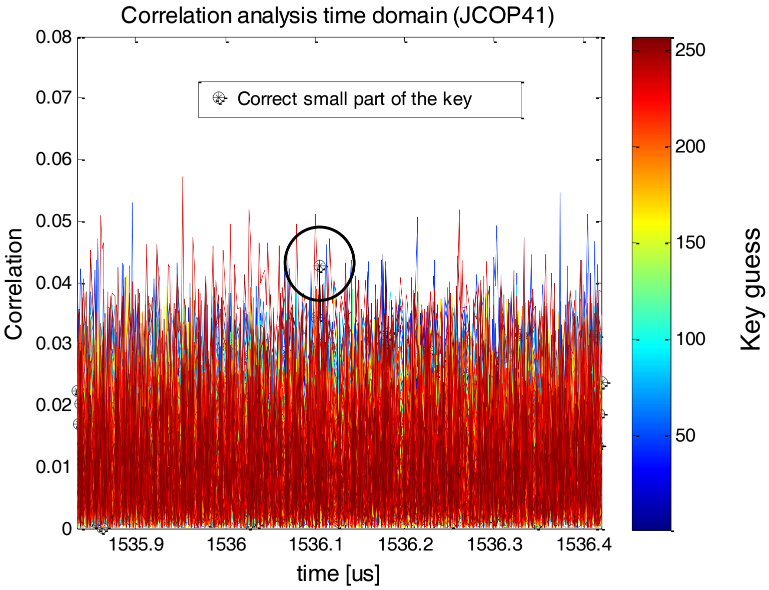


Figure 7.19: Projection from the correlation analysis matrix in the time domain showing correlation vs. time, using 5,888 traces.

The difference between the correlation of key guess 234 and key guess 45 ranked in second place is 4.81%. Figure 7.20 shows the highest correlation values obtained in the correlation matrix for each key guess. The other keys tested (0x9E and 0xA2) using the JCOP41 were unsuccessful in recovering the correct key.

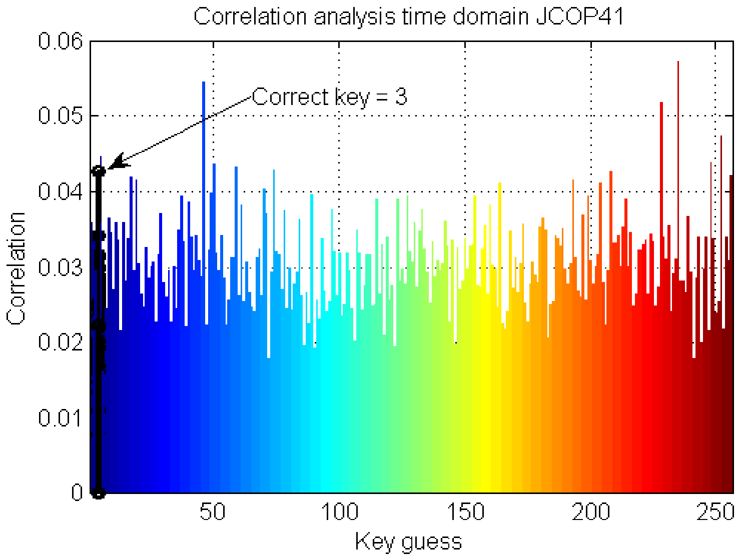


Figure 7.20: Projection of the correlation matrix in the time domain showing correlation vs. key guess using 5,888 traces.

Using SPA and considering as a reference the hypothetical power consumption model from the correct key used for processing the plaintexts, the average of the EM traces sorted according to their Hamming weight is displayed in Figure 7.21. In the upper part of the figure, the 8 clock cycles processed are displayed and in the bottom part, the figure zooms in on 1,536.07 µs, the time where the correct key used presented its maximum correlation value. At this point, the sequence of the Hamming weights from top to bottom is 1, 2, 3, 4, 0, 7, 5, 6, and 8. The values 0 and 7 seem to be out of sequence.

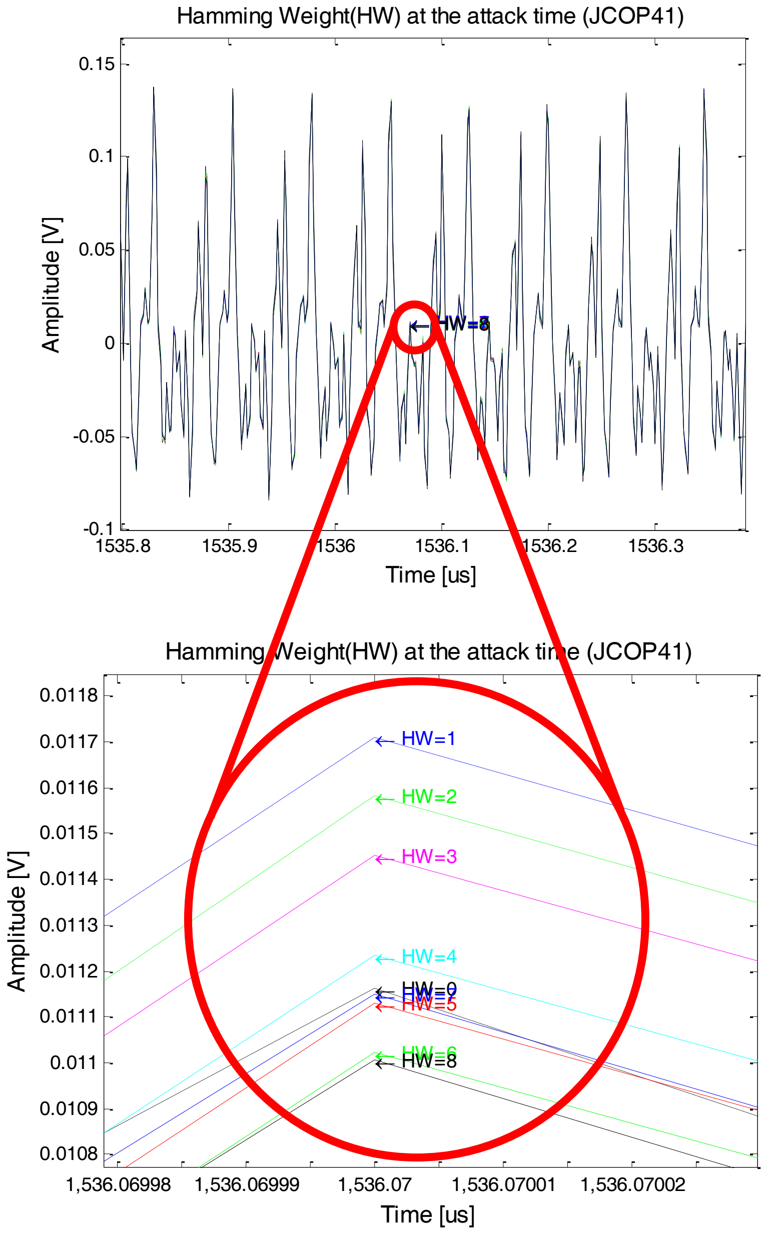


Figure 7.21 Averages of the EM traces according to the Hamming weight of the correct key guess power model.

7.3.2 Frequency Domain Analysis Using the JCOP41

For the frequency domain analysis, the EM traces used in Section 7.3.1 are transformed into the frequency domain using the FFT. The correlation matrix values obtained after analysing the corresponding 5,888 power spectrum traces are represented in Figure 7.22. The values corresponding to the correct key guess (0x03) are displayed in black with circles on top.

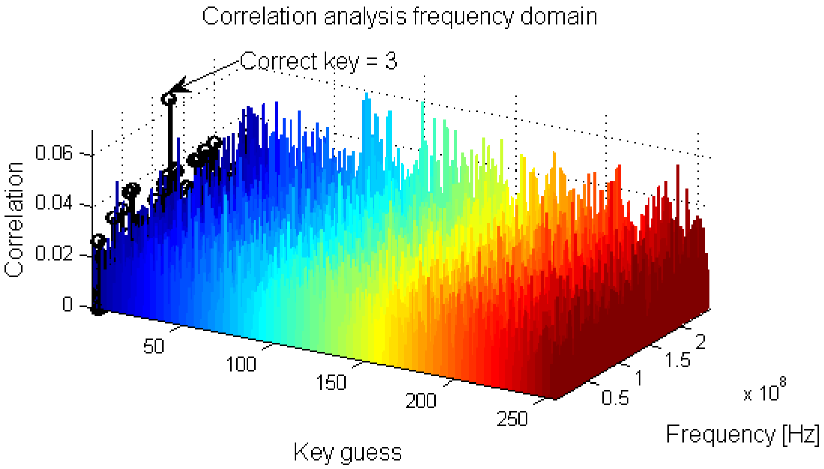


Figure 7.22: Representation of the correlation matrix in the frequency domain using 5,888 traces.

A projection of the correlation matrix across the different frequencies is shown in Figure 7.23. The maximum value 0.06627 occurs at 116.44 MHz and the maximum value is 5.31 standard deviations bigger than the mean of the other correlations at that particular frequency. The difference between the first and second biggest correlation (key guess 0x6A) is 2.97%.

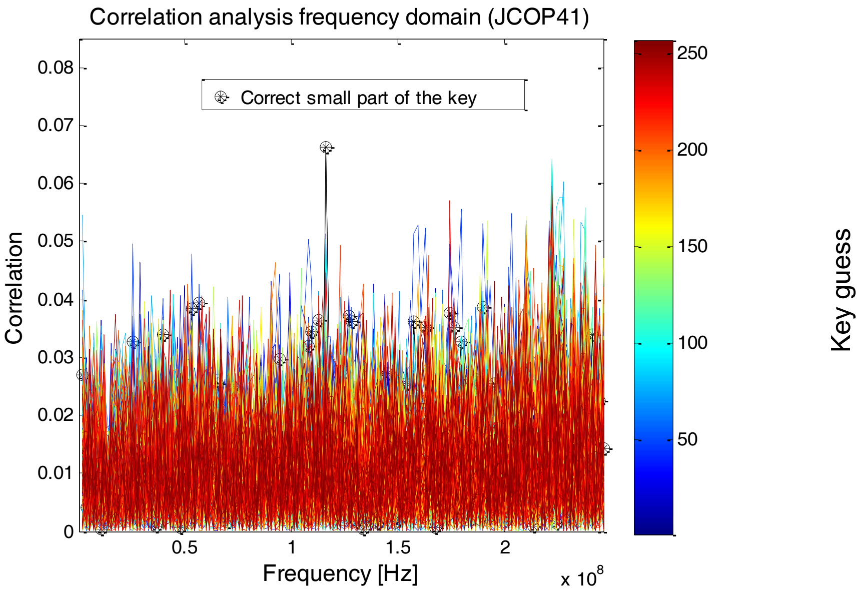


Figure 7.23: Projection of the correlation matrix after using 5,888 traces.

The highest correlations for each key guess are represented in Figure 7.24. In the graph the correct key displayed in black is 3.58 standard deviations bigger than the average of all other maximums for the correct key.

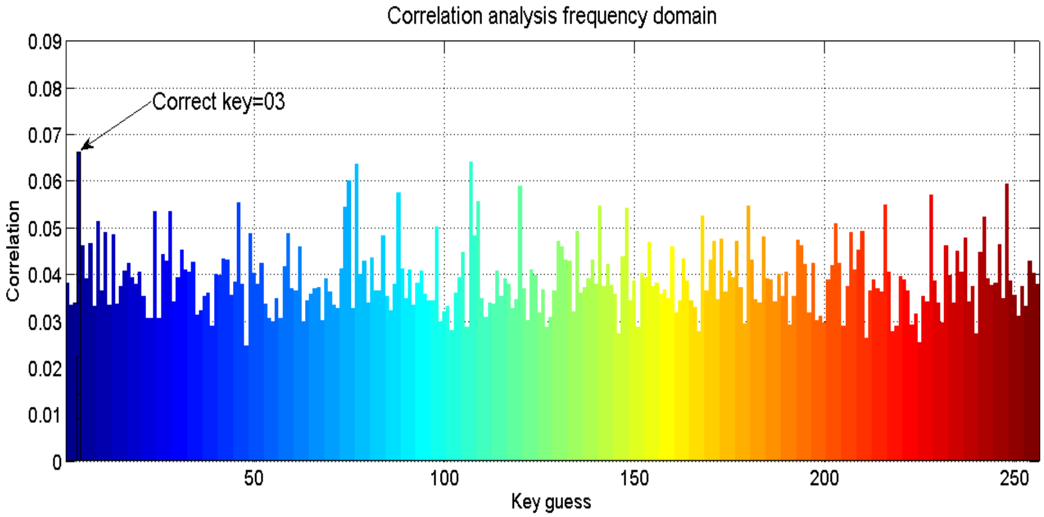


Figure 7.24: Projection of the correlation analysis matrix for the JCOP41 using correlation analysis in the frequency domain and 5,888 power spectrum traces.

The average of all power spectrums according to their Hamming weight using the hypothetical power models with the correct key are displayed in Figure 7.25. In the upper part of the figure, the average power spectrum corresponding to each Hamming weight is represented. The frequencies related to the harmonics of the card reader clock frequency (13.56 MHz) reaching the maximum power at 41 MHz are visible with higher magnitudes. The figure zooms in on the 116.44 MHz frequency, This is the point where the correlation matrix returned the greatest value. In the case of the other key guesses tested, the results are presented in Table 7.5.

Table 7.5: Test results for other keys used JCOP41

Key used	Frequency [MHz]	Rank	Max correlation correct key
0x03	116	1	0.0662
0x9E	68	7	0.0402
0xA2	116	2	0.0468

The sequence of Hamming weights from smallest to largest at this point is 0, 1, 2, 3, 4, 6, 5, 7, and 8. The only value that looks out of sequence is HW=6. Converting the values from dB to watts at 116.44 MHz, the difference in the power measured using the GMR probe between HW=0 and HW=1 is 0.45 μW and between HW=7 and HW=8 is 1.63 μW.

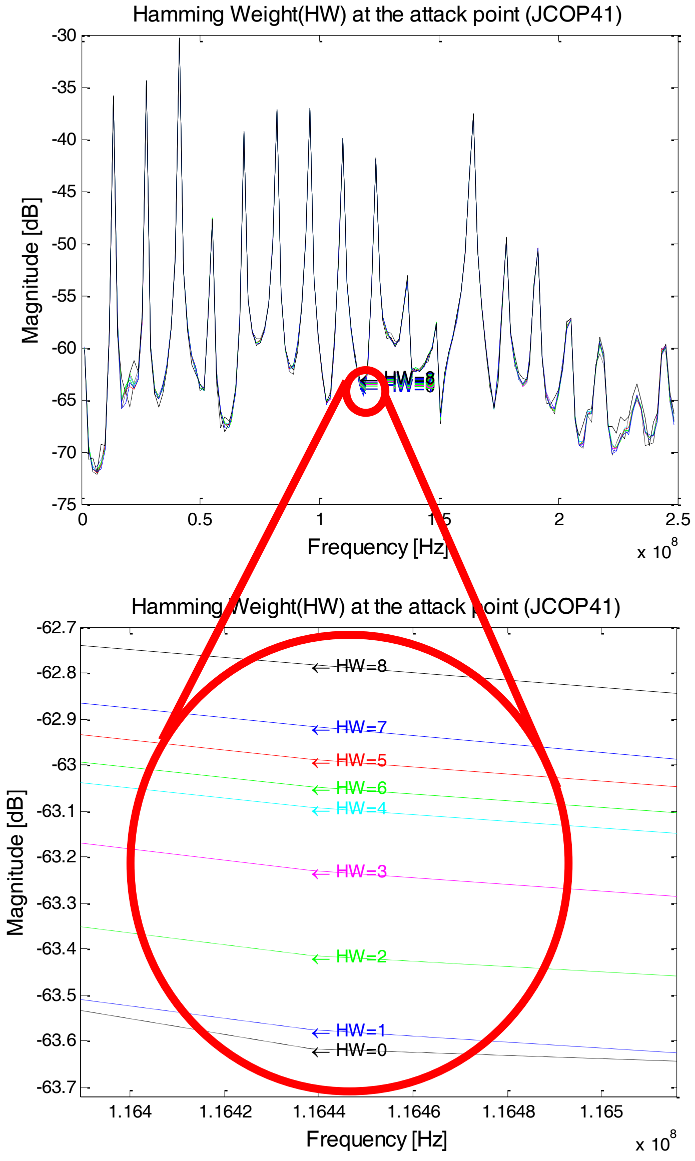


Figure 7.25: Average Hamming weight using the hypothetical power model of the correct key.

7.3.3 Measuring the Effectiveness of the Analysis in the Time and Frequency Domain

The performance of correlation analysis in the time and frequency domain are compared by contrasting the accuracy results obtained for different number of traces using each technique. The accuracy results obtained after analysing 1 to 5,888 EM traces and their corresponding power spectrum traces are displayed in Figure 7.26, where it is visible that the time domain analysis, represented by a blue dashed line, did not returned the correct key (up to 5,888 traces analysed), while for the frequency domain analysis with 5,101 traces, it was possible to recover the correct key. The result of adding more traces are visible in the upper part of the graph where a linear scale was used. These variations do not go below 0.0078 and after 5,501 traces, the minimum is 0.125 but it returns to 1.

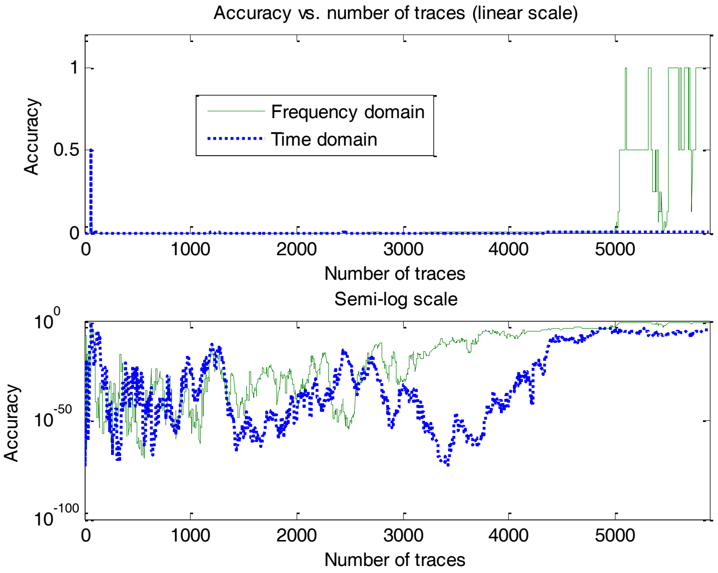


Figure 7.26: Contrast between time and frequency domain analysis using accuracy as the metric (JCOP41).

7.4 Comparison to Previous Research and Summary

Unlike previous research where the attacks on Java Cards have used power analysis (Vermoen et al. 2007, Sterckx, 2009), in this research EM analysis is used. In a recent attack on contactless smart cards published by (Oswald and Paar, 2011), DESFire cards, a different type of card from the Java Cards analyzed in this research, were used.

The DESFire cards do not have garbage collection and the previously published attacks are on a hardware implementation. In this research, the Java Cards have garbage collection and the AES software implementations under analysis or attack require significantly longer execution times.

In this chapter, the three illustrative cases where correlation analysis is used to analyse the EM power traces captured on two types of Java Cards are presented.

In the first case, the card used was the JCOP30 where the IC chip was removed from the card and placed inside a Faraday cage. The number of traces analysed are 39,936 with a sampling rate of 250 MS/s. Using correlation analysis in the time domain and frequency domain, it was possible to determine the small part of the key used when the card runs a software implementation of AES.

In the second case studied, the card is the JCOP30 without modifications and the analysis uses 5,888 EM traces, acquired with a sampling resolution of 500 MS/s. For these traces, both forms of analysis (in time and frequency domains) were able to return the correct key. It is clear in Figure 7.17 that the tendency of the correct key is well defined in returning an accuracy of 1.

The third case presented, corresponds to a JCOP41 card without modifications. For this analysis, 5,888 traces were acquired with a sampling rate of 500 MS/s. Using correlation analysis in the frequency domain, it was possible to recover the correct small part of the key. An incorrect key was returned using correlation analysis in the time domain.

A summary of the results from this chapter is presented in Table 7.6. For the three cases studied, the analysis in the frequency domain returned higher correlations than the time domain. An element to consider in the analysis, beside the magnitude of the correlation, is the distance Δ measured between the highest correlation value from correct key guess and the average of all maximums for the other key guesses. These distances are bigger in the case of the frequency domain analysis. To illustrate the meaning of these distances, let us say that if the values of the maximum correlation for all guesses were normally distributed, only 1 of 256 values is expected to be farther than Δ=2.886 standard deviations above the mean; 1 of 370 is expected to be farther than Δ=3, and 1 of 5,222 is expected to be farther than Δ=3.73.

Table 7.6: Summary of results for the correlation analysis presented.

Type of card	Number of traces	Sampling rate	Time domain		Frequency domain	
			Maximum Correlation for the correct key	Time point attacked	Maximum Correlation for the correct key	Frequency attacked
JCOP30 modified in Faraday cage	39,936	250 MS/s	0.01959 Δ=2.98	1,498.65 μs	0.02565 Δ=3.29	112.85 MHz
JCOP30 no modifications	5,888	500 MS/s	0.05407 Δ=3.27	1,498.92 μs	0.07016 Δ=3.73	170.06 MHz
JCOP41 no modifications	5,888	500 MS/s	0.04266 * Δ=1.9	1,536.1 μs	0.06627 Δ=3.58	116.44 MHz

*unable to recover the correct key with 5,888 traces.

The correlation results presented in this chapter maybe do not show the "high" spikes reported in Chapter 3 and Chapter 5. However, with the exception of the analysis to the JCOP41 in the time domain, the results are above the noise floor calculated with Equation (3-12) which is 0.02 for 39,936 traces and 0.0521 for 5,888 traces. Moreover, the maximum correlations found are bigger than those estimated for distinguishing a peak using Equation (3-11) with an error probability α=0.995 and the respective number of traces S. This means that the attacks should be considered meaningful.

Chapter 8

Discussions and Conclusions

Studying the Java Cards working in contactless mode required the analysis of complex EM emissions from a system on chip (SoC) device in the presence of a strong external EM field generated by the card reader. This challenge motivated research in experimental setup and sophisticated signal analysis. A number of features in the final setup were unique to the contactless cards, such as removing the card reader from its box and placing it in a coordinates table where the position of the card and EM sensors can be easily adjusted. Other adjustments like the replacement of the power supply to reduce the noise on the acquired signal and the change from an inductive probe to a GMR probe as described in Chapter 5 are critical for the successful analysis. Besides the quality of the EM acquisitions, the long time interval acquired for the study of Java Cards is another challenge. In the case of the attack on the microcontroller (Chapter 3), the time of the attack after the trigger is 7.35 µs and in the DPA contest (Chapter 4), the 10 rounds of AES take 0.65 µs. For the Java Card the 10 rounds of AES take 3 seconds and the points of the attack that are reported in this work occur around 1500 µs after the trigger. Additionally, the Java Cards contain a garbage collection utility that may interrupt the execution of the program at any time causing further large misalignments as described in Section 6.1.3.

Although calculating a correlation is based on simple mathematical principles, part of the complexity of the side channel resides in the magnitude of the sets to associate, and the challenges to acquire analyzable data. An alternate approach could attempt to acquire "many" power or EM traces during the time the system is encrypting or decrypting the data. Quantifying the value of "many" is not trivial and may require an iterative process to gain better knowledge about the system under study. Before this research, the number of traces required to test or attack a software implementation of AES running on Java Cards was unknown. It takes around 3 seconds to complete a full encryption of AES 128 using a Java software implementation on a JCOP30. Using a sampling rate of 500 MS/s each trace would contain 1,500,000,000 samples. If the number of traces to analyse were set to 40,000 traces, it becomes a computational challenge to correlate one matrix of size [40,000×1,500,000,000] corresponding to the power EM traces with another of [40,000×256] corresponding to the power model (40,000 plaintexts assuming 256 small parts of the key). The size of the power traces matrix, assuming 2 bytes per sample, would be 120 Tbytes and the analysis in the time domain would require correlating 384 billion vectors of size 40,000. This research determined the time when the smart card is processing one of the S-boxes in the JCOP30 and JCOP41. Then, with less than 5,888 traces the small parts of the key used for encrypting information using the Java Card were retrieved using correlation analysis.

The analysis of three different experimental setups using Java Cards and the applet test 2 are presented in Chapter 7. Two of them involve a JCOP30 and one uses the JCOP41; in the first case, the IC was removed from the card and placed 30 cm away from the card reader inside a Faraday cage; in the other two cases, the Java Cards were analyzed without any modification. In the first case, reducing the intensity of the card reader by acquiring the EM signals away from it and having a sampling rate of 250 MS/s helped to identify a possible time of attack by analyzing longer time intervals without using higher resolutions. The other two Java Card analyses used a 500 MS/s sampling rate. Although the analysis for the Java Cards did not return those high spikes reported in the microcontroller analysis (Chapters 3 and 5), the use of correlation analysis in the frequency domain allowed one to recover the correct small part of the key in the three cases. Additionally, correlation analysis in the time domain was successful in recovering the correct key for the two setups that used the JCOP30; however, analyzing the JCOP41 in the time domain failed to recover the correct key guess considering up to 5,888 traces.

Comparing the presented analyses among the Java Card set-ups, the analyses in the frequency domain returned better results than correlation analysis in the time domain. The analyses of JCOP30 without modifications using 5,888 traces and a sampling rate of 500 MS/s returned higher correlations and bigger distances between the maximum correlation and the mean from all maximums than the other two setups analyzed (JCOP30 Faraday cage and JCOP41). The results for the JCOP30 using correlation analysis in the frequency domain returned a maximum correlation *r*=0.07016. It was 9.79% bigger than the second highest correlation and 3.73 standard deviations bigger than the mean of all maximums. Using correlation analysis in the time domain, the maximum correlation was *r*=0.05407, only 2.5% bigger than the second highest key guess and the distance to the mean of all maximums was 3.27 standard deviations. The analysis of the JCOP30 where the chip was removed from the card also returned the correct key guess but the use of a lower resolution (250 MS/s) seems to have affected the outcome, because the distance between the maximum correlation and the average of all means was smaller than in the case where 500 MS/s were used. The analysis of the JCOP41 in the frequency domain was successful in returning the correct key, although the maximum correlation was smaller than for JCOP30. The time domain analysis reached a maximum correlation for the correct key guess of 0.04266, 1.9 standard deviations above the mean of all maximums for all keys.

This research has highlighted the importance that a few frequencies are more prone to leak information about the secret data processed. A methodology called fast correlation in the frequency domain, described in Chapter 2, analyzes a selected range of frequencies from the power spectrum and correlates them with the hypothetical power model. This type of analysis has been used successfully in retrieving the correct key guess in different systems such as microcontrollers (Chapters 3 and 5), FPGAs (Chapter 4), and Java Cards (Chapter 7). This analysis also helps to return the correct key guess in the presence of small misalignments on the order of 20 ns (Chapter 3). Using fast correlation analysis in the frequency domain, approximately 40 EM power traces were needed from the microcontroller to recover the correct key guess. In the case of the power traces from the DPACv2 public database (key 31), around 5,000 traces were required to recover the whole 128 bits of the cipher key and for the JCOP30 without modifications 2,437 traces were needed.

The use of a reduced range of frequencies not only helps to speed up the processing time but also seems to require a fewer number of traces as shown in Chapter 3 and Chapter 7. An important concern with this method is the selection of the leaking frequencies. Some ways to select these frequencies have been presented in

(Mateos and Gebotys, 2010). The use of a reduced number of frequencies for performing differential frequency analysis was applied in (Lu et al. 2009a). More recently in (Oswald and Paar, 2011), a selected range of frequencies was used for studying the security of the DESFire smart cards using correlation analysis; however, in this case the authors use custom hardware real-time filters to acquire the EM traces and later, they analysed those acquisitions in the time and frequency domain.

The difficulties of acquiring EM traces from the Java Cards that successfully retrieve the correct key using a commercial inductive EM probe motivated a search for alternative probes. After, unsuccessfully testing several probes that included inductors of different sizes, wire gauges, number of loops, even VCR heads for VHS tapes, a probe that uses giant magnetoresistance sensors was proposed (Mateos and Gebotys, 2011). This probe has shown to be reliable and significantly helped to investigate the side channel. First, it was used to acquire EM traces that allowed retrieving the correct key guess from a microcontroller (Chapter 5) and later from the JCOP30 and JCOP41 Java Cards (Chapter 7).

As previously discussed, having to analyse traces acquired using different EM probes, at different positions, at different sampling rates, with a variety of cipher keys, and over a very large analysis time window requires extensive computational processing time. To overcome this issue, a number of programs that run correlation analysis in the time and in the frequency domain with optimised execution time were developed. One version of these programs based on fast correlation in the frequency domain was submitted to the DPACv2, since one of the evaluation criteria was the execution time. The results from the DPACv2 indicate that this proposed attack was the fastest attack and among the fastest attacks it is the one that had the best success rate at 20,000 traces (Chapter 4).

Similar to the DPA contest where different attacks are compared, side channel research needs to evaluate the effectiveness of different attacks and/or data. Usually obtaining the correct key is a good gauge, but having the certainty that the results are robust is even better. In this research two metrics, accuracy and estimation are presented. Accuracy quantifies the correctness of an attack by sorting all the key guesses from more likely to less likely and using an exponential scale to assign a value depending on how close the attack is from reaching the correct key guess. For example, an attack that recovers the correct key is evaluated two times better than one which returns the correct key in second place and four times better than another one that returns the correct key in third place. This exponential scale helps to minimise the weight of those attacks that did not return the correct key among the top positions. This scale requires one to know the correct key to evaluate the performance of the attack. However, when the attacker does not know the correct key beforehand, estimation can be used to evaluate the outcome from a correlation attack. This metric qualifies the dispersion between the maximum value for the correlation and the mean for all maximums. If the distance is bigger than δ standard deviations, the attack is considered meaningful, otherwise, it is not. These two metrics have been used to evaluate the effectiveness of the different attacks presented in this research.

There are no previous reports of EM side channel attacks on Java Cards working contactlessly, apart from an attack on a DESFire card (Oswald and Paar, 2011). Table 8.1 presents a summary of the characteristics of both works. However, it is important to highlight that the subjects of study are different, specifically cards with different types of algorithm implementations executing on different hardware (processor vs custom hardware) and different numbers of traces considered in the analysis.

The proposed attacks have successfully returned the correct small part of the key used by a software implementation of AES. Although, the experiment is unrealistic in that the same value is written multiple times into memory, it provided a good method for identifying the attack region. This supported analysis of leakage that could be exploited by an attacker.

In the first experiment reported in Chapter 7, a total of 39,936 traces were used and the IC chip was removed from the card to identify a possible time of attack. Once the time of the attack was known, it was possible to move forward and in the following experiment an unmodified card was used with acquisitions of fewer traces. In this case the 5,888 traces were the maximum number of traces that one could acquire with the oscilloscope using a sampling rate of 500 MS/s and a window time of 10 μs.

Table 8.1: Characteristics of the analysis studies to the Java Card and the DESFire Card.

Characteristic:	Side Channel Analysis of a Java-based Contactless Smart Card	(Oswald and Paar, 2011)
Type of card	Java Card	DESfire
Model of card	JCOP30 and JCOP41	MF3ICD40
Mode of operation during the attack	Contactless	Contactless
Algorithm attacked	Software implementation AES	Hardware implementation (triple-)DES
Average running time of the cryptographic algorithm	3 s	8.2 μs
EM probe used	GMR	Inductive probe
Additional hardware used for processing signal	none	Analog demodulator (custom hardware)
Methodology of attack used	Correlation analysis time and frequency domain	Correlation analysis time and frequency domain
Sampling rate	500 MS/s	500 MS/s
Window of time used in the analysis or Duration of 1 EM trace	20 acquisitions of 10 μs (200 μs). For frequency domain were analyzed windows of 0.589 μs	1.5 μs
Correlation at the point of attack (time domain)	0.054 at 5,888 traces (JCOP30) 0.042 at 5,888 traces (JCOP41)	0.014 at 500,000 traces
Correlation at the point of attack (frequency domain)	0.070 at 5,888 traces (JCOP30) 0.066 at 5,888 traces (JCOP41)	0.02 at 500,000 traces

In the case of the smart cards used, little information is available and this makes it difficult to know the status of registers and buses in a given time. The writing of a value of 0 to a register before the attacked value allows one to apply the Hamming weight model instead of the Hamming distance, whereas explained in Chapters 3 and 6 an attacker generally needs to know the previous state of the register or bus under study. The proximity to the card reader and its strong magnetic field is another constraint because it reduces the signal to noise ratio (SNR). Besides, any effect that the strong EM field could induce on the emissions from the smart cards, from acquiring EM traces closer to the card reader, usually requires bigger voltage scales in the oscilloscope. The use of higher scales implies bigger quantization noise and lower SNR (Johns and Martin, 1997). Considering that for correlation analysis the correlation factor ρ and SNR (for small SNRs) are related by $\rho^2 \sim SNR$ (Mangard et al. 2007), a reduction in the SNR implies smaller correlation values. Additionally, the SNR is inversely proportional to the number of traces n required to identify the correct key, $n \approx \frac{1}{SNR}$ (Mangard et al. 2007). Consequently, one may assume that working closer to the card reader will negatively affect the SNR and one shall expect a reduction in the correlation magnitudes and an increase in the number of traces required for a successful attack.

8.1 Summary of Contributions

In this thesis, a complete methodology for analysing the side channel from Java Cards while executing a cryptographic algorithm in software in contactless mode is presented. The analyses described can be applied, with the corresponding adjustments, to other types of smart cards and electronic devices that use cryptographic algorithms. The main contributions presented in this work are listed next.

- The first EM side channel attack of a commercial smart card JCOP30 and JCOP41 running in contactless mode and executing a software implementation of part of AES
- The fastest processing time per trace attack, submitted to the DPA contest version 2.
- First application of GMR sensors to side channel attacks.
- A new type of side channel attack, called fast correlation in the frequency domain.
- Two new metrics to evaluate the effectiveness of the side channel analysis.
- A complete methodology for the side channel analysis of the smart cards.
- A novel setup for Java Card analysis that includes a co-ordinates analysis table and a programmable instruction based trigger, for the oscilloscope.

8.2 Future Work

With the results obtained in this research, new experiments and potential research paths arise; some of them are listed in this section. The only impractical aspect of the proposed attack is the fact that 20 writes of the AES state were inserted into the partial AES application running on the smart card. Hence, it would be interesting in the future to attempt to remove this constraint. In particular, future research could study further the practicality of this attack by reducing the number of writes to the register and at the same time increasing the number of traces.

Side channel attacks on the S-boxes of different Java Cards running a software implementation of AES were presented in Chapter 7. It would also be suitable to test this methodology with a card that runs a hardware implementation of AES.

The EM traces acquired to analyze the Java Cards were acquired using the GMR probe without additional filters or analog demodulators. It would be interesting to study the effects of such hardware on the analyses presented here.

Some constraints from the actual oscilloscope were mentioned in Chapter 6. It would be convenient to evaluate the performance of other oscilloscopes or analog to digital converters with more than 8-bit resolution and larger memory storage. In particular, oscilloscopes which are not limited by scope memory, such as those which can automatically acquire and transfer traces. For example, oscilloscopes which can be programmed to transfer traces out of the oscilloscope memory and continue to acquire traces automatically.

To improve the security of a system, it may be important to hide or mask those frequencies that leak more information, hence, research into this area of efficient countermeasure design would be important.

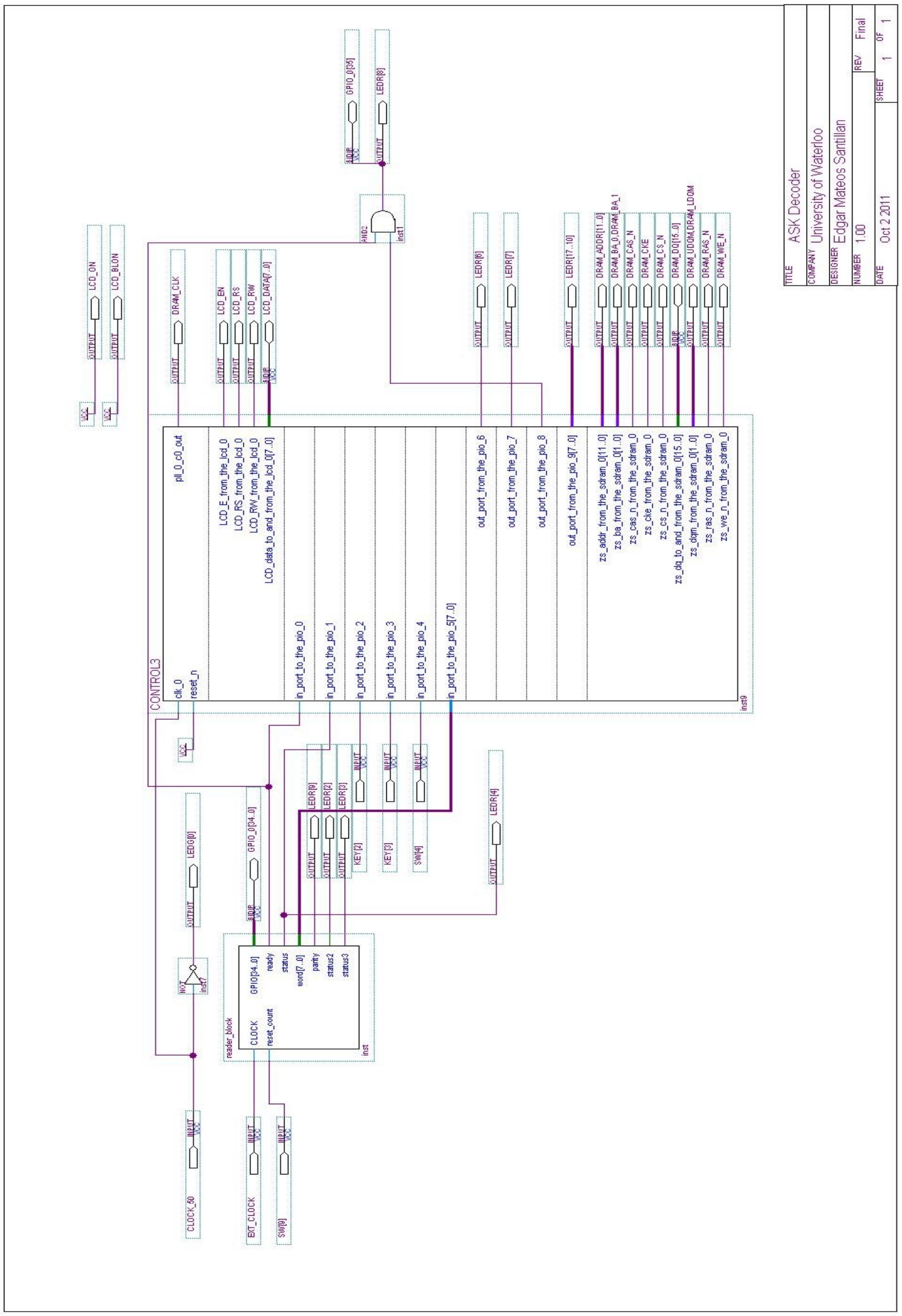
Appendix

A.1 Java Code that Implements Applet Test 2 (S-box 21 Iterations)

```
/* This program implements the Applet test 2 (S-box 21 iterations)
* described in Algorithm 6.2.
* A Java Card receives a plaintext within an APDU command and uses it to
* calculate the result of S-box(plaintext xor key), where key is a small
* part of the cipher key stored in the Java Card.
* The result of the operation is written 21 times to a variable and the
* result is sent to the card reader.
*/
package opera;
import javacard.framework.APDU;
import javacard.framework.Applet;
import javacard.framework.JCSystem;
public class OperaApp extends Applet {
       byte value[] = JCSystem.makeTransientByteArray((short)1,
JCSystem.CLEAR_ON_DESELECT);
       // A small part of the cipher key is set
       private static final byte key =(byte)0xA2;
       // AES S-box
       private static final byte keySub[] = {
              (byte) 0x63,(byte) 0x7C,(byte) 0x77,(byte) 0x7B,
              (byte) 0xF2,(byte) 0x6B,(byte) 0x6F,(byte) 0xC5,
              (byte) 0x30,(byte) 0x01,(byte) 0x67,(byte) 0x2B,
              (byte) 0xFE,(byte) 0xD7,(byte) 0xAB,(byte) 0x76,
              (byte) 0xCA,(byte) 0x82,(byte) 0xC9,(byte) 0x7D,
              (byte) 0xFA,(byte) 0x59,(byte) 0x47,(byte) 0xF0,
              (byte) 0xAD,(byte) 0xD4,(byte) 0xA2,(byte) 0xAF,
              (byte) 0x9C,(byte) 0xA4,(byte) 0x72,(byte) 0xC0,
              (byte) 0xB7,(byte) 0xFD,(byte) 0x93,(byte) 0x26,
              (byte) 0x36,(byte) 0x3F,(byte) 0xF7,(byte) 0xCC,
              (byte) 0x34,(byte) 0xA5,(byte) 0xE5,(byte) 0xF1,
              (byte) 0x71,(byte) 0xD8,(byte) 0x31,(byte) 0x15,
              (byte) 0x04,(byte) 0xC7,(byte) 0x23,(byte) 0xC3,
              (byte) 0x18,(byte) 0x96,(byte) 0x05,(byte) 0x9A,
              (byte) 0x07,(byte) 0x12,(byte) 0x80,(byte) 0xE2,
              (byte) 0xEB,(byte) 0x27,(byte) 0xB2,(byte) 0x75,
              (byte) 0x09,(byte) 0x83,(byte) 0x2C,(byte) 0x1A,
              (byte) 0x1B,(byte) 0x6E,(byte) 0x5A,(byte) 0xA0,
              (byte) 0x52,(byte) 0x3B,(byte) 0xD6,(byte) 0xB3,
              (byte) 0x29,(byte) 0xE3,(byte) 0x2F,(byte) 0x84,
              (byte) 0x53,(byte) 0xD1,(byte) 0x00,(byte) 0xED,
              (byte) 0x20,(byte) 0xFC,(byte) 0xB1,(byte) 0x5B,
              (byte) 0x6A,(byte) 0xCB,(byte) 0xBE,(byte) 0x39,
              (byte) 0x4A,(byte) 0x4C,(byte) 0x58,(byte) 0xCF,
              (byte) 0xD0,(byte) 0xEF,(byte) 0xAA,(byte) 0xFB,
              (byte) 0x43,(byte) 0x4D,(byte) 0x33,(byte) 0x85,
              (byte) 0x45,(byte) 0xF9,(byte) 0x02,(byte) 0x7F,
              (byte) 0x50,(byte) 0x3C,(byte) 0x9F,(byte) 0xA8,
              (byte) 0x51,(byte) 0xA3,(byte) 0x40,(byte) 0x8F,
              (byte) 0x92,(byte) 0x9D,(byte) 0x38,(byte) 0xF5,
              (byte) 0xBC,(byte) 0xB6,(byte) 0xDA,(byte) 0x21,
                (byte) 0x10,(byte) 0xFF,(byte) 0xF3,(byte) 0xD2,
                (byte) 0xCD,(byte) 0x0C,(byte) 0x13,(byte) 0xEC,
                (byte) 0x5F,(byte) 0x97,(byte) 0x44,(byte) 0x17,
                (byte) 0xC4,(byte) 0xA7,(byte) 0x7E,(byte) 0x3D,
                (byte) 0x64,(byte) 0x5D,(byte) 0x19,(byte) 0x73,
                (byte) 0x60,(byte) 0x81,(byte) 0x4F,(byte) 0xDC,
                (byte) 0x22,(byte) 0x2A,(byte) 0x90,(byte) 0x88,
                (byte) 0x46,(byte) 0xEE,(byte) 0xB8,(byte) 0x14,
                (byte) 0xDE,(byte) 0x5E,(byte) 0x0B,(byte) 0xDB,
                (byte) 0xE0,(byte) 0x32,(byte) 0x3A,(byte) 0x0A,
                (byte) 0x49,(byte) 0x06,(byte) 0x24,(byte) 0x5C,
                (byte) 0xC2,(byte) 0xD3,(byte) 0xAC,(byte) 0x62,
                (byte) 0x91,(byte) 0x95,(byte) 0xE4,(byte) 0x79,
                (byte) 0xE7,(byte) 0xC8,(byte) 0x37,(byte) 0x6D,
                (byte) 0x8D,(byte) 0xD5,(byte) 0x4E,(byte) 0xA9,
                (byte) 0x6C,(byte) 0x56,(byte) 0xF4,(byte) 0xEA,
                (byte) 0x65,(byte) 0x7A,(byte) 0xAE,(byte) 0x08,
                (byte) 0xBA,(byte) 0x78,(byte) 0x25,(byte) 0x2E,
                (byte) 0x1C,(byte) 0xA6,(byte) 0xB4,(byte) 0xC6,
                (byte) 0xE8,(byte) 0xDD,(byte) 0x74,(byte) 0x1F,
                (byte) 0x4B,(byte) 0xBD,(byte) 0x8B,(byte) 0x8A,
                (byte) 0x70,(byte) 0x3E,(byte) 0xB5,(byte) 0x66,
                (byte) 0x48,(byte) 0x03,(byte) 0xF6,(byte) 0x0E,
                (byte) 0x61,(byte) 0x35,(byte) 0x57,(byte) 0xB9,
                (byte) 0x86,(byte) 0xC1,(byte) 0x1D,(byte) 0x9E,
                (byte) 0xE1,(byte) 0xF8,(byte) 0x98,(byte) 0x11,
                (byte) 0x69,(byte) 0xD9,(byte) 0x8E,(byte) 0x94,
                (byte) 0x9B,(byte) 0x1E,(byte) 0x87,(byte) 0xE9,
                (byte) 0xCE,(byte) 0x55,(byte) 0x28,(byte) 0xDF,
                (byte) 0x8C,(byte) 0xA1,(byte) 0x89,(byte) 0x0D,
                (byte) 0xBF,(byte) 0xE6,(byte) 0x42,(byte) 0x68,
                (byte) 0x41,(byte) 0x99,(byte) 0x2D,(byte) 0x0F,
                (byte) 0xB0,(byte) 0x54,(byte) 0xBB,(byte) 0x16};

public static void install(byte[] bArray, short bOffset, byte bLength) {
                // GP-compliant Java Card applet registration
                new OperaApp()
                .register(bArray, (short) (bOffset + 1), bArray[bOffset]);
}
        public void process(APDU apdu) {
                // Good practice: Return 9000 on SELECT
                if (selectingApplet()) {
                        return;
                }
                // Read the APDU command
                byte[] buf = apdu.getBuffer();
                // Initialize a variable where the result of the s-box
                // operation will be stored
                byte v=0;
                // Compute Value=SubByte(P xor k)
                value[0]= (byte)(keySub[(int)(buf[2]^key)&0xff]);
                // Stores the result of the S-box in to the variable for
                // 21 times
                // To avoid any overhead related with "For" loops and
                // conditional operations the values of v are explicitly
                .// assigned
                v=value[0] ; //1
                // For the Apple test 1 (s-box one iteration) here starts
                // the response
                // For the Apple test 2 (s-box 21 iterations) 20 cycles of
                // clearing and writing are perform
                v=0;
                v=value[0] ; //2
                v=0;
                v=value[0] ; //3
                v=0;
                v=value[0] ; //4
                v=0;
                v=value[0] ; //5
                v=0;
                v=value[0] ; //6
                v=0;
                v=value[0] ; //7
                v=0;
                v=value[0] ; //8
                v=0;
                v=value[0] ; //9
                v=0;
                v=value[0] ; //10
                v=0;
                v=value[0] ; //11
                v=0;
                v=value[0] ; //12
                v=0;
                v=value[0] ; //13
                v=0;
                v=value[0] ; //14
                v=0;
                v=value[0] ; //15
                v=0;
                v=value[0] ; //16
                v=0;
                v=value[0] ; //17
                v=0;
                v=value[0] ; //18
                v=0;
                v=value[0] ; //19
                v=0;
                v=value[0] ; //20
                v=0;
                v=value[0] ; //21
                // Sends the result of the s-box operation to the card
                // reader
                // (The following 3 lines were commented when measuring
                // the times in table 6.1)
                apdu.setOutgoing();
                apdu.setOutgoingLength((short)1);
                apdu.sendBytesLong(value,(short) 0,(short) 1);
            }
        }
```

A.2 ASK Decoder



References

AERTS, W., DE MULDER, E., PRENEEL, B., VANDENBOSCH, G.A.E. and VERBAUWHEDE, I., 2006. Matching shielded loops for cryptographic analysis, *European Conference on Antennas and Propagation: EuCAP 2006,* 6-10 Nov. 2006, European Space Agency.

AGRAWAL, D., ARCHAMBEAULT, B., RAO, J.R. and ROHATGI, P., 2002. The EM side-channel(s), *4th International Workshop Revised Papers*, 13-15 Aug. 2002, Springer-Verlag, pp. 29-45.

ANONYMOUS, 2009, JCOP41 without AES?, Oracle discussion forums, Java Card. [Homepage of Oracle], [Online]. Available:
https://forums.oracle.com/forums/thread.jspa?threadID=1749683 [Jan. 8, 2012].

ATMEL, 2007, AT89C51RD2, AT89C51ED2. 8-bit Flash Microcontroller [Homepage of Atmel Corporation], [Online]. Available:
http://www.atmel.com/dyn/resources/prod_documents/doc4235.pdf [Jan. 8, 2012].

BAER, M., DREXLER, H. and PULKUS, J., 2010. Improved Template Attacks, *COSADE 2010 – First International Workshop on Constructive Side–Channel Analysis and Secure Design*, 4-5 Feb. 2010, pp. 81.

BERKES, J.E., 2008. *Side-channel monitoring of contactless java cards*, University of Waterloo.

BEVAN, R. and KNUDSEN, E., 2003. Ways to enhance differential power analysis, *Information Security and Cryptology - ICISC 2002. 5th International Conference. Revised Papers*, 28-29 Nov. 2002, Springer-Verlag, pp. 327-42.

BOAK, D., 1973. *A history of U.S. communications security (U).* 1 edn. Maryland, USA: National Security Agency.

BOEY, K.H., LU, Y., O'NEILL, M. and WOODS, R., 2010. Differential Power Analysis of CAST-128, *2010 IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2010)*, 5-7 Jul. 2010, IEEE Computer Society, pp. 143-8.

BRIER, E., CLAVIER, C. and OLIVIER, F., 2004. Correlation power analysis with a leakage model, *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2004. 6th International Workshop.*, 11-13 Aug. 2004, Springer-Verlag, pp. 16-29.

BULENS, P., DANGER, J.L., DUC, G., ELAABID, A., FLAMENT, F., GUILLEY, S., HOMMA, N., HOOGVORST, P., MEYNARD, O., PAUGET, F., SAUVAGE, L., STANDAERT, F.X. and CHARVILLON, N.V., Feb. 25, 2011, 2011, DPA Contest v2 2009/2010. [Online] Available: http://www.dpacontest.org/v2/index.php [Jan. 8, 2012].

CARLUCCIO, D., LEMKE, K. and PAAR, C., 2005. Electromagnetic side channel analysis of a contactless smart card: First results, *Proceedings of Workshop on RFID and Lightweight Crypto*, 14 Jul. 2005.

CHARI, S., RAO, J.R. and ROHATGI, P., 2002. Template attacks, *4th International Workshop Revised Papers*, 13-15 Aug. 2002, Springer-Verlag, pp. 13-28.

CHEN, K., ZHAO, Q., ZHANG, P. and DENG, G., 2008. The power of electromagnetic analysis on embedded cryptographic ICs, *2008 International Conference on Embedded Software and Systems Symposia (ICESS Symposia)*, 29-31 Jul. 2008, IEEE, pp. 197-201.

CHEN, Z., 2000. *Java Card technology for Smart Cards: architecture and programmer's guide.* Boston: Addison-Wesley.

CLAVIER, C., CORON, J. and DABBOUS, N., 2000. Differential power analysis in the presence of hardware countermeasures, *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2000. Second International Workshop*, 17-18 Aug. 2000, Springer-Verlag, pp. 252-63.

CORON, J., 1999. Resistance against differential power analysis for elliptic curve cryptosystems, *First International Workshop, CHES'99*, 12-13 Aug. 1999, Springer-Verlag, pp. 292-302.

DOGET, J., PROUFF, E., RIVAIN, M. and STANDAERT, F.X., 2011. Univariate Side Channel Attacks and Leakage Modeling, *COSADE 2011 – Second International Workshop on Constructive Side–Channel Analysis and Secure Design*, 24-25 Feb. 2011, pp. 1-15.

ELECTRO-METRICS INC., ed, 2004. *Instruction Manual: Near Field Probe Set Broadband Response Model EM- 6992.*

FINKENZELLER, K., 2003. *RFID handbook: fundamentals and applications in contactless smart cards and identification.* 2 edn. Chichester, England ; Hoboken, N.J.: Wiley.

FREEDMAN, D., PISANI, R. and PURVES, R., 1998. *Statistics.* 3 edn. New York: W.W. Norton.

GANDOLFI, K., MOURTEL, C. and OLIVIER, F., 2001. Electromagnetic analysis: concrete results, *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2001. Third International Workshop*, 14-16 May 2001, Springer-Verlag, pp. 251-61.

GEBOTYS, C.H., HO, S. and TIU, C.C., 2005. EM analysis of Rijndael and ECC on a wireless Java-based PDA, *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2005. 7th International Workshop*, 29 Aug.-1 Sept. 2005, Springer-Verlag, pp. 250-64.

GEBOTYS, C.H., 2010. *Security in embedded devices.* New York ; London: Springer.

GEBOTYS, C.H., 2006. A split-mask countermeasure for low-energy secure embedded systems. *ACM Trans.Embed.Comput.Syst.,* **5**(3), pp. 577-612.

GIERLICHS, B., BATINA, L., TUYLS, P. and PRENEEL, B., 2008. Mutual information analysis: a generic side-channel distinguisher, *10th International Workshop*, 10-13 Aug. 2008, Springer-Verlag, pp. 426-42.

HENDRY, M., 2007. *Multi-application smart cards: technology and applications.* Cambridge ; New York: Cambridge University Press.

HODGERS, P., BOEY, K.H. and O'NEILL, M., 2011. Power spectral density side channel attack overlapping window method, *(DSD 2011)*, 31 Aug.-2 Sept. 2011, IEEE Computer Society, pp. 274-8.

IBM, 2002. *JCOP The IBM GlobalPlatform JavaCardTM implementation.* IBM.

IBM, 2000, *JCOP30 Technical Brief* [Homepage of International Business Machines Corp], [Online]. Available: ftp://ftp.software.ibm.com/software/pervasive/info/JCOP30Brief.pdf [Jan. 8, 2012].

IEC, 2002. *61967-1, Integrated circuits - Measurement of electromagnetic emissions, 150 kHz to 1 GHz - Part 1: General conditions and definitions.* Draft 47A/632/FDIS edn. Geneva, Switzerland: International Electrotechnical Commission.

IMJELA, R., 1996. *Overview of Products and Application Notes for Smartcards, Aplication note AN96006.* Philips Semiconductors.

ISO/IEC, 2011. *7816-1, Identification cards — Integrated circuit cards — Part 1: Cards with contacts — Physical characteristics.* 2nd edn. Geneva, Switzerland: International Organization for Standardization, International Electrotechnical Commission.

ISO/IEC, 1999a. *14443-2, Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 2: Radio frequency power and signal interface.* Draft FCD 14443-2 edn. Geneva, Switzerland: International Organization for Standardization; International Electrotechnical Commission.

ISO/IEC, 1999b. *14443-3, Identification cards — Contactless integrated circuit(s) cards — Proximity cards — Part 3: Initialization and anticollision.* Draft FCD 14443-3 edn. Geneva, Switzerland: International Organization for Standardization; International Electrotechnical Commission.

JOHNS, D. and MARTIN, K.W., 1997. *Analog integrated circuit design.* New York: John Wiley & Sons.

JUN, W.J., 2003, Smart Card Technology Capabilities. Available:
http://csrc.nist.gov/publications/nistir/IR-7056/Capabilities/Jun-SmartCardTech.pdf
[Jan. 8, 2012].

KASAP, S.O., 2006. *Principles of electronic materials and devices.* 3 edn. Boston: McGraw-Hill.

KASPER, T., OSWALD, D. and PAAR, C., 2011. Side-Channel Analysis of Cryptographic RFIDs with Analog Demodulation, *Proceedings of RFIDSec 2011*, Springer LNCS.

KASPER, T., CARLUCCIO, D. and PAAR, C., 2007. An embedded system for practical security analysis of contactless smartcards, *Proceedings of Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems. First IFIP TC6/WG8.8/WG 11.2 International Workshop, WISTP 2007*, 9-11 May 2007, Springer, pp. 150-60.

KASPER, T., OSWALD, D. and PAAR, C., 2009. EM Side-Channel Attacks on Commercial Contactless Smartcards Using Low-Cost Equipment, *10th International Workshop, WISA 2009*, 25-27 Aug. 2009, Springer-Verlag, pp. 79-93.

KASPER, T., OSWALD, D. and PAAR, C., 2011. Wireless security threats: Eavesdropping and detecting of active RFIDs and remote controls in the wild, *19th International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2011,* 15-17 Sep. 2011, IEEE Computer Society, pp. 435-40.

KEAN, H.B., HODGERS, P., LU, Y., O'NEILL, M. and WOODS, R., 2010. Security of AES Sbox designs to power analysis, *2010 17th IEEE International Conference on Electronics, Circuits and Systems (ICECS 2010)*, 12-15 Dec. 2010, IEEE, pp. 1232-5.

KEIL, 1997. *MCB251 Evaluation Board.* 1 edn. Keil Software.

KIZHVATOV, I., 2009. Side channel analysis of AVR XMEGA crypto engine, *Embedded Systems Week 2009, ESWEEK 2009 - 4th Workshop on Embedded Systems Security, WESS 2009,* 15 Oct 2009, Association for Computing Machinery.

KOCHER, P., JAFFE, J. and JUN, B., 1999. Differential power analysis, *Proceedings of Advances in Cryptology - CRYPTO'99. 19th Annual International Cryptology Conference*, 15-19 Aug. 1999, Springer-Verlag, pp. 388-97.

KOCHER, P., LEE, R., MCGRAW, G., RAGHUNATHAN, A. and RAVI, S., 2004. Security as a new dimension in embedded system design, *Design Automation Conference*, 7-11 June 2004, ACM, pp. 753-60.

KOCHER, P.C., 1996. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems, *Advances in Cryptology - CRYPTO '96*, 18-22 Aug. 1996, Springer-Verlag, pp. 104-13.

KOCHER, P., JAFFE, J., JUN, B. and ROHATGI, P., 2011. *Introduction to differential power analysis.* Springer Berlin / Heidelberg.

KONING, G., HOEPMAN, J.-. and GARCIA, F.D., 2008. A practical attack on the MIFARE classic, *2 International Conference, CARDIS 2008*, 8-11 Sept. 2008, Springer-Verlag, pp. 267-82.

KOPF, B. and BASIN, D., 2007. An information-theoretic model for adaptive side-channel attacks, *14th ACM Conference on Computer and Communications Security, CCS'07,* 29 Oct.-2 Nov. 2007, Association for Computing Machinery, pp. 286-96.

KRESALEK, V., SMOLA, M. and KOSINA, T., 2008. Scanning of electromagnetic radiation for EMC and data security purposes, *42nd Annual 2008 IEEE International Carnahan Conference on Security Technology*, 13-16 Oct. 2008, IEEE, pp. 117-20.

LE, T., CANOVAS, C. and CLEDIERE, J., 2008. An overview of side channel analysis attacks, *2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS '08, March 18, 2008 - March 20*, 2008, Association for Computing Machinery, pp. 33-43.

LU, Y., BOEY, K.H., O'NEILL, M. and MCCANNY, J.V., 2009a. Practical Comparison of Differential Power Analysis Techniques on an ASIC Implementation of the AES Algorithm, *IET Irish Signals and Systems Conference (ISSC 2009)*, 10-11 June 2009, IET, pp. 6.

LU, Y., BOEY, K.H., O'NEILL, M., MCCANNY, J.V. and SATOH, A., 2009b. Is the differential frequency-based attack effective against random delay insertion? *SiPS 2009*, 7-9 Oct. 2009, IEEE, pp. 51-6.

MALLINSON, J.C., 2002. *Magneto-resistive and spin valve heads: fundamentals and applications.* 2 edn. San Diego: Academic Press.

MANGARD, S., 2004. Hardware countermeasures against DPA - a statistical analysis of their effectiveness, *Topics in Cryptology - CT-RSA 2004. Cryptographers' Track at the RSA Conference 2004*, 23-27 Feb. 2004, Springer-Verlag, pp. 222-35.

MANGARD, S., 2003. Exploiting Radiated Emissions - EM Attacks on Cryptographic ICs, L. OSTERMANN, ed. In: *Proceedings of Austrochip 2003*, pp. 13.

MANGARD, S., OSWALD, E. and POPP, T., Apr. 16, 2010, 2010a, DPAbook.org Power analysis attacks- revealing the secrets of smart cards [Homepage of Graz University of Technology], [Online]. Available: http://www.dpabook.org/ [Jan. 8, 2010].

MANGARD, S., OSWALD, E. and POPP, T., 2010b, Power Analysis Attacks - Revealing the Secrets of Smartcards. Matlab Workspaces, Workspace 3 (WP3). [Online] Available: http://www.dpabook.org/onlinematerial/matlabws/index.htm [Jan. 8, 2012].

MANGARD, S., OSWALD, E. and POPP, T., 2007. *Power analysis attacks: revealing the secrets of smart cards.* New York: Springer.

MASSEY, J.L., 1994. Guessing and entropy, *Proceedings of the 1994 IEEE International Symposium on Information Theory,* 27 Jun.-1 Jul. 1994, IEEE, pp. 204.

MATEOS, E. and GEBOTYS, C.H., 2011. Side channel analysis using giant magneto-resistive (GMR) sensors, *COSADE 2011 – Second International Workshop on Constructive Side–Channel Analysis and Secure Design*, 2011, pp. 42-49.

MATEOS, E. and GEBOTYS, C.H., 2010. A new correlation frequency analysis of the side channel, *5th Workshop on Embedded Systems Security, WESS '10,* 24 Oct 2010, ACM, pp. 4:1-4:8.

MAYER-SOMMER, R., 2000. Smartly analyzing the simplicity and the power of simple power analysis on smartcards, *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2000. Second International Workshop*, 17-18 Aug. 2000, Springer-Verlag, pp. 78-92.

MEDWED, M. and OSWALD, E., 2008. Template attacks on ECDSA, *9th International Workshop, WISA 2008*, 23-25 Sept. 2008, Springer, pp. 14-27.

MENEZES, A.J., VAN OORSCHOT, P.C. and VANSTONE, S.A., 1997. *Handbook of applied cryptography.* Boca Raton: CRC Press.

MESSERGES, T.S., 2000. Using second-order power analysis to attack DPA resistant software, *Proceedings of Cryptographic Hardware and Embedded Systems - CHES 2000. Second International Workshop*, 17-18 Aug. 2000, Springer-Verlag, pp. 238-51.

MESSERGES, T.S., DABBISH, E.A. and SLOAN, R.H., 2002. Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers,* **51**(5), pp. 541-52.

MESSERGES, T.S., DABBISH, E.A. and SLOAN, R.H., 1999. Power analysis attacks of modular exponentiation in smartcards, *First International Workshop, CHES'99*, 12-13 Aug. 1999, Springer-Verlag, pp. 144-57.

MIFARE, 2011, Security of MF3ICD40. [Online]. Available:
http://mifare.net/technology/security/mifare-desfire-d40/ [Jan. 8, 2012].

MIFARE, Nov. 10, 2008, 2008, Information for system integrators. [Online]. Available:
http://www.mifare.net/technology/security/mifare-classic/information-for-system-integrators/ [Jan. 8, 2012].

MORADI, A., MOUSAVI, N., PAAR, C. and SALMASIZADEH, M., 2009. A Comparative Study of Mutual Information Analysis under a Gaussian Assumption, *10th International Workshop, WISA 2009*, 25-27 Aug. 2009, Springer-Verlag, pp. 193-205.

NATIONAL SECURITY AGENCY and CENTRAL SECURITY SERVICE, 1972. *TEMPEST: A Signal Problem.* [Online]. Available:
http://www.nsa.gov/public_info/_files/cryptologic_spectrum/tempest.pdf [Jan. 8, 2012]..

NIST, 2001. *Announcing the Advanced Encryption Standard (AES).* National Institute of Standards and Technology (NIST).

NOHL, K. and PLOTZ, H., 2007. MIFARE, Little Security, Despite Obscurity, *24th Chaos Communication Congress*, 28 Dec. 2007.

NOVAK, R., 2002. SPA-Based Adaptive Chosen-Ciphertext Attack on RSA Implementation, *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems: Public Key Cryptography* 2002, Springer-Verlag, pp. 252-62.

NVE, a, AA and AB-Series Analog Sensors [Homepage of Non Volatile Electronics Corporation], [Online]. Available:
http://www.nve.com/Downloads/analog_catalog.pdf [Jan. 8, 2012].

NVE, b, Sensor Engineering and Application Notes [Homepage of Non Volatile Electronics Corporation], [Online]. Available:
http://www.nve.com/Downloads/apps.pdf [Jan. 8, 2012].

NXP, 2009a, AN10841 MIFARE Plus Card Coil Design [Homepage of NXP semiconductors], [Online]. Available:
http://www.nxp.com/documents/application_note/AN10841_1.pdf [Jan. 8, 2012].

NXP, 2009b, CLRC632 Multiple protocol contactless reader IC (MIFARE/I-CODE1) [Homepage of NXP Semiconductors], [Online]. Available:
http://www.nxp.com/documents/data_sheet/CLRC632.pdf [Jan. 8, 2012].

NXP, 2009c, Smart solutions for smart services [Homepage of NXP Semiconductors], [Online]. Available: http://www.nxp.com/documents/line_card/75016728.pdf [Jan 8, 2012].

NXP, Jan. 24, 2008, 2008, P5Cx012/02x/40/73/80/144 family Secure dual interface and contact PKI smart card controller [Homepage of NXP Semiconductors], [Online]. Available:
http://www.nxp.com/documents/data_sheet/P5CX012_02X_40_73_80_144_FAM_SDS.pdf [Jan. 8, 2012].

OMNIKEY, 2005, CardMan 5121PC-Linked 13.56 MHz RFID Smart Card Reader [Homepage of Txsystems], [Online]. Available:
http://www.txsystems.com/downloads/CardMan_5121_Datasheet_E.pdf
[Jan. 8, 2012].

ORS, S.B., GURKAYNAK, F., OSWALD, E. and PRENEEL, B., 2004. Power-analysis attack on an ASIC AES implementation, *International Conference on Information Technology: Coding and Computing*, 5-7 Apr. 2004, IEEE Comput. Soc, pp. 546-52.

OSWALD, D. and PAAR, C., 2011. *Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World.* Springer Berlin / Heidelberg.

PARAB, J.S., SHELAKE, V.G., KAMAT, R.K., NAIK, G.M. and SPRINGERLINK, 2007. *Exploring C for Microcontrollers.* Dordrecht: Springer Science+Business Media B.V.

PEARSON, J. and ALBUS, Z., 2009. *The Right Kind of Microcontroller for Contactless Smart ICs.* [Online]. Available: http://www.ti.com/ww/in/newsletters.html
[Jan. 8, 2012].

PENG, Z., GAOMING, D., QIANG, Z. and KAIYAN, C., 2009. EM Frequency Domain Correlation Analysis on Cipher Chips, *2009 1st International Conference on Information Science and Engineering (ICISE 2009)*, 26-28 Dec. 2009, IEEE, pp. 1729-32.

PHILIPS, 2006, JCOP41/72B4 V2.2.1 on Secure Triple Interface PKI Smart Card Controller [Homepage of Philips Semiconductors], [Online]. Available: http://www.cardid.ro/datasheet/NXP/JCOP41_SPI.pdf [Jan. 8, 2012].

PHILIPS, 2004, *P5CD072 Secure Dual Interface PKI smart card controller*. Available: http://www.datasheetarchive.com/P5CD072*%C2-datasheet.html [Jan. 8, 2012].

PHILIPS, 2003, Mifare proX P8RF5016 Secure Dual Interface Smart Card IC, short form specification [Homepage of Philips Semiconductors], [Online]. Available: http://www.classic.nxp.com/acrobat_download2/other/identification/sfs051814.pdf [Jan. 8, 2012].

PLOS, T., HUTTER, M. and HERBST, C., 2008. Enhancing Side-Channel Analysis with Low-Cost Shielding Techniques, CHRISTOPH LACKNER, TIMM OSTERMANN, MICHAEL SAMS, RONALD SPILKA, ed. In: *Proceedings of Austrochip 2008, Linz, Austria,* 8 Oct. 2008*,* pp. 90.

PROUFF, E. and RIVAIN, M., 2010. Theoretical and practical aspects of mutual information-based side channel analysis. *International Journal of Applied Cryptography,* **2**(2), pp. 121-38.

QUISQUATER, J. and SAMYDE, D., 2001. *ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards.* Springer-Verlag.

R&S, 2011. *R&S SMA100A Operating Manual.* Version 9 edn. Munich, Germany: ROHDE & SCHWARZ GmbH & Co. KG.

RANKL, W., 2007. *Smart card applications: design models for using and programming smart cards.* Chichester, England ; Hoboken, NJ: John Wiley & Sons Ltd.

RANKL, W. and EFFING, W., 2000. *Smart card handbook.* 2 edn. Chichester, England ; New York: Wiley.

RCIS, 2010, Side-channel Attack Standard Evaluation Board (SASEBO) [Homepage of Research Center for Information Security], [Online]. Available: http://www.rcis.aist.go.jp/special/SASEBO/SASEBO-GII-en.html [Jan. 8, 2012].

RISCURE, 2011, Inspector. Available: http://www.riscure.com/tools/inspector [Jan. 8, 2012].

ROHATGI, P., 2006. Side-Channel Attacks. In: H. BIDGOLI, ed, *Handbook of information security: Treats, vulnerabilities, prevention, detection and management.* Hoboken, N.J.: John Wiley & Sons, pp. 241-59.

ROUSSEAU, L., 2012, Smartcard_list [Homepage of Ludovic Rousseau], [Online]. Available: http://ludovic.rousseau.free.fr/softwares/pcsc-tools/smartcard_list.txt [Jan. 8, 2012].

SCHIMMEL, O., DUPLYS, P., BOEHL, E., HAYEK, J., BOSCH, R. and ROSENSTIEL, W., 2010. Correlation power analysis in frequency domain, *COSADE 2010 - First International Workshop on Constructive Side-Channel Analysis and Secure Design*, 4-5 Feb. 2010.

STANDAERT, F.-., MALKIN, T.G. and YUNG, M., 2009. A unified framework for the analysis of side-channel key recovery attacks, *28th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 26-30 Apr. 2009, Springer-Verlag, pp. 443-61.

STANDAERT, F.-., ORS, S.B., QUISQUATER, J.-. and PRENEEL, B., 2004. Power analysis attacks against FPGA implementations of the DES, *Proceedings of Field-Programmable Logic and Applications. 14th International Conference, FPL 2004.*, 30 Aug.-1 Sept. 2004, Springer-Verlag, pp. 84-94.

STERCKX, M., 2009. *Implementation and Side-Channel Analysis of Anonymous Credentials on Java Card Platforms*, Katholieke Universiteit Leuven.

STOREY, B., 2002, Computing Fourier Series and Power Spectrum with MATLAB [Homepage of Franklin W. Olin College of Engineering], [Online]. Available: http://faculty.olin.edu/bstorey/Notes/Fourier.pdf [Jan. 8, 2012].

SVENDA, P., 2009, Java Card algorithms support test [Homepage of Faculty of Informatics Masaryk University], [Online]. Available: http://www.fi.muni.cz/~xsvenda/docs/JCOP41_112008.pdf [Jan. 8, 2012].

TEKTRONIX, 2003. *CSA7000 Series, TDS7000 Series, & TDS6000 Series Instruments User Manual. 071-7010-03.* Version 2.3.0 edn. Beaverton, OR: Tektronix, Inc.

THANH-HA LE, CLEDIERE, J., CANOVAS, C., ROBISSON, B., SERVIERE, C. and LACOUME, J.-., 2006. A proposition for correlation power analysis enhancement, *Proceedings of Cryptographic Hardware and Embedded Systems-CHES 2006. 8th International Workshop*, 10-13 Oct. 2006, Springer, pp. 174-86.

VERMOEN, D., WITTEMAN, M. and GAYDADJIEV, G.N., 2007. Reverse engineering Java card applets using power analysis, *Proceedings of Information Security Theory and*

Practices. Smart Cards, Mobile and Ubiquitous Computing Systems. First IFIP TC6/WG8.8/WG 11.2 International Workshop, WISTP 2007. , 9-11 May 2007, Springer, pp. 138-49.

VEYRAT-CHARVILLON, N. and STANDAERT, F.-., 2009. Mutual information analysis: how, when and why? *11th International Workshop*, 6-9 Sept. 2009, Springer-Verlag, pp. 429-43.

WHITNALL, C. and OSWALD, E., 2011. A comprehensive evaluation of mutual information analysis using a fair evaluation framework, *31st Annual International Cryptology Conference, CRYPTO 2011,* 14-18 Aug. 2011, Springer Verlag, pp. 316-34.

YAMAGUCHI, M., TORIDUKA, H., KOBAYASHI, S., SUGAWARA, T., HOMMAA, N., SATOH, A. and AOKI, T., 2010. Development of an on-chip micro shielded-loop probe to evaluate performance of magnetic film to protect a cryptographic LSI from electromagnetic analysis, *2010 IEEE International Symposium on Electromagnetic Compatibility - EMC 2010*, 25-30 Jul. 2010, IEEE, pp. 103-8.

YOUNG, H.D. and FREEDMAN, R.A., 1996. *University physics.* 9th edn. Reading, Mass.: Addison-Wesley.