S-BOX CLASSIFICATION AND SELECTION IN SYMMETRIC-KEY ALGORITHMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HACI ALİ ŞAHİN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

SEPTEMBER 2016

Approval of the thesis:

S-BOX CLASSIFICATION AND SELECTION IN SYMMETRIC-KEY ALGORITHMS

submitted by **HACI ALİ ŞAHİN** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** _____

Prof. Dr. Ferruh Özbudak
Head of Department, **Cryptography** _____

Prof. Dr. Ferruh Özbudak
Supervisor, **Cryptography, METU** _____

Dr. Begül Bilgin
Co-supervisor, **COSIC, KU Leuven** _____

Examining Committee Members:

Prof. Dr. Ferruh Özbudak
Cryptography, METU _____

Assoc. Prof. Dr. Sedat Akleylek
Department of Computer Engineering, Samsun Ondokuz Mayıs University _____

Assoc. Prof. Dr. Murat Cenk
Cryptography, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: HACI ALİ ŞAHİN

Signature :

ABSTRACT

S-BOX CLASSIFICATION AND SELECTION IN SYMMETRIC-KEY ALGORITHMS

ŞAHİN, Hacı Ali
M.S., Department of Cryptography
Supervisor : Prof. Dr. Ferruh Özbudak
Co-Supervisor : Dr. Begül Bilgin

September 2016, 37 pages

Side-channel analysis(SCA) attack is one of popular attacks which is mainly interested in environmental effects of embedded devices such as power, time and temperature. Since power analysis attack is one of the most efficient SCA attacks and it poses a threat for current cryptographic algorithms, we are mainly interested in certain applications of it, namely differential and correlation power analysis. Also, we share our observations on behaviours of attacks for different cases. The number of bits used in model and the number of traces are critical parameters which determine efficiencies of attacks. In our experiment, we see that the number of bits used in model and the number of traces are directly proportional to efficiencies of algorithms. However, they also increase time complexities of algorithms. Therefore, these should be determined in terms of algorithms, the quality of traces and computer that runs algorithm. Moreover, s-boxes are the main component of symmetric-key algorithms which provide resistance against cryptanalysis methods. Main focus of studies in s-boxes is the classification of $n \times n$ s-boxes for $n \in \{3, 4, 5, 6, 7, 8\}$. Affine equivalence provides researchers with classifying all s-boxes and understanding influences of s-boxes into cryptography. However, current technology and classification algorithms limit us to classify permutations only for small n values. Until now, only 4×4 permutations are classified and listed completely by algorithms generated by De Cannière [22]. Selection of s-boxes by considering only cryptanalysis is not enough to generate cryptographic algorithms. Implementation of cryptographic algorithms on embedded devices can cause leakages

and may not simply provide secure system. Therefore, new countermeasure methods may need to be applied into algorithm to prevent embedded devices from leakages. However, these methods may cause high complexities and can damage performances of algorithms because of gates used in s-box implementations. Therefore, we aim to classify 5×5 permutations and to analyse each class in terms of their cryptographic properties. 5×5 quadratic permutations are classified into 75 class and properties of all quadratic classes are listed in terms of critical properties.

Keywords : Classification of 5x5 quadratic permutations, linear cryptanalysis, differential cryptanalysis, side channel analysis, differential power analysis, correlation power analysis

ÖZ

S KUTULARINDA SINIFLANDIRMA VE SIMETRIK ALGORITMALARDA S KUTUSU SEÇIMI

ŞAHİN, Hacı Ali
Yüksek Lisans, Kriptoloji
Tez Yöneticisi : Prof. Ferruh Özbudak
Ortak Tez Yöneticisi : Dr. Begül Bilgin

Eylül 2016, 37 sayfa

Yan kanal analizi, güç tüketimi, güvenlik algoritmasının aldığı süre veya dışarıya verdiği ısı gibi gömülü cihazların girdi ve çıktılarından sonra etrafa verdiği etkileri inceleyen atak çeşididir. Günümüzdeki popülerliği ile beraber etkin kullanımı birçok kriptografik algoritması için tehdit unsuru olmaktadır. Bu nedenle, farklı parametrelerin ataklar üzerindeki etkisini görmek için AES-128 güvenlik algoritması üzerinde bir kaç test yapmak tez süreci boyunca sahip olduğum hedeflerden bir tanesidir. Ataklarda bulunan modellerin kullandığı bit sayısı ve güç tüketim tablo sayısı atakların başarısını ve etkinliğini etkileyen iki kritik unsur olarak görünmektedir. Fakat algoritmaların çalışma sürelerindeki artışı bu parametrelerinin artışı ile doğru orantılıdır. Bu nedenle güç tüketim tablo sayısı ve kullanılan bit sayısı atağı kullanan kişi tarafından atak algoritması ve bilgisayara göre belirlenmelidir. Ayrıca, özellikle simetrik anahtar kullanan algoritmalarda kullanılan s kutularının seçimi tüm kriptografik ataklar için önemlidir. Bu nedenle, s kutuları farklı açılardan birçok çalışma ile incelenmiştir. Özellikle, $3, 4, 5, 6, 7$ ve 8 boyutlu uzaylar üzerinde tanımlı s kutularının sınıflandırılması s kutu seçimi için çok kritik rol oynamaktadır. Fakat günümüz teknolojisi ve geliştirilen algoritmalar [22] sadece 4 boyutlu uzaylar için sınıflandırmayı mümkün kılmaktadır. S kutularının farklı boyutlarda sınıflandırılması için henüz bir gelişme olmaması ve s kutu seçimi için sınıflandırma yönteminin faydası bizim bu alanda çalışmamız için ilham kaynağı olmuştur. 5 boyutlu uzaylarda sınıfların yüksek sayısı düşünüldüğünde ikinci dereceden s kutularının sınıflandırılması sağlanmıştır. İkinci dereceden 5 boyutlu s kutuları 75 farklı sınıftan oluşmaktadır. Bulunan sınıfların en önemli özelliği sahip

olduğu s kutularının kriptografik açıdan aynı özellikleri bulunmasıdır. Bu sayede sadece sınıfları oluşturmak değil, ayrıca oluşturulan sınıfları kriptografik açıdan değerlendirme ve listeleme şansımız da bulunmaktadır.

Anahtar Kelimeler: Liner ve differensiyel ataklar, yan kanal analizi, fark güç analizi, korelasyon güç analizi, permutasyonların sınıflandırılması

To My Darling and Family

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my thesis supervisor Prof. Ferruh Ozbudak for his support and valuable advice. It is a great pleasure to thank co-supervisor Dr. Begül Bilgin for her support. This study would not be complete without her guidance and comments.

I am also so grateful to Dusan BOZILOV. Almost all valuable contributions in this thesis is completed by Dusan and Dr. Begül.

I am also thankful to my thesis defence committee members for their useful comments and discussions

I want to express my genuine appreciation and warm thanks for my father Nurettin, my mother Filiz, my sister Fatma and my love Ceren. I always feel their endless support during these studies.

Finally, I would like to thank The Scientific and Technological Research Council of Turkey (TUBİTAK) for their support.

TABLE OF CONTENTS

ABSTRACT vii

ÖZ ix

ACKNOWLEDGMENTS xiii

TABLE OF CONTENTS xv

LIST OF FIGURES xvii

LIST OF TABLES xix

LIST OF ABBREVIATIONS xxi

CHAPTERS

1 INTRODUCTION 1

1.0.1 Asymmetric-key Cryptography 1

1.0.2 Symmetric-key Algorithms 2

1.0.3 About the Thesis 4

2 SIDE CHANNEL ANALYSIS 5

2.1 Notations 7

2.2 Power-Monitoring Attacks 8

2.2.1 Differential Power Analysis 10

2.2.2 Correlation Power Analysis 12

2.3 Contribution 13

2.4 Countermeasures 15

3 S-BOXES CLASSIFICATION 17

3.1 Preliminaries 17

3.1.1 Finding Linear Representative 18

3.2 Contribution 19

3.2.1 New representation of permutations 19

3.2.2 Construction of S_{ANF} for 5 $\times$ 5 quadratic permutations 20

3.2.3 Main Algorithm 22

3.2.4 Equivalence Classes of 5 x 5 quadratic permutations 23

3.2.5 Multiplicative Complexity 24

4 CONCLUSION AND FUTURE WORKS 25

REFERENCES 27

APPENDICES

A DPA and CPA Attack Results 31

B Classification of Quadratic 5 x 5 S-Boxes 37

LIST OF FIGURES

Figure 1.1 Classification of Cryptographic Algorithms [18] 2

Figure 2.1 Cryptographic Device 5

Figure 2.2 Classification of Attacks 6

Figure 2.3 CMOS Inverter 7

Figure 2.4 Power consumption Trace of AES-128 14

Figure 2.5 Effects of Number of bits used by Model in DPA-Hamming Distance Attacks 15

Figure A.1 DPA Attack into First Key with 1000 traces 31

Figure A.2 DPA Attack into First Key with 2000 traces 32

Figure A.3 DPA Attack into First Key with 3000 traces 32

Figure A.4 DPA Attack into First Key with 4000 traces 33

Figure A.5 DPA Attack into First Key with 10000 traces 33

Figure A.6 CPA Attack into First Key with 10000 traces- Number of Bits used by Model : 1 34

Figure A.7 CPA Attack into First Key with 10000 traces- Number of Bits used by Model : 2 34

Figure A.8 CPA Attack into First Key with 10000 traces- Number of Bits used by Model : 4 35

LIST OF TABLES

Table 3.1 Linearity of Classes 23
Table 3.2 Differential uniformity of Classes 23
Table 3.3 Multiplicative Complexity of 5x5 quadratic Permutation Classes . . 24

Table B.1 Quadratic Classes of quadratix 5×5 permutations 37
Table B.2 Quadratic Classes of quadratix 5×5 permutations 38

LIST OF ABBREVIATIONS

SNR	Signal to Noise Ratio
DDT	Difference Distribution Table
LAT	Linear Approximation Table
DPA	Differential Power Analysis
CPA	Correlation Power Analysis
CMOS	Complementary Metal–Oxide–Semiconductor
LSB	Least Significant Bit
GUI	Graphical User Interface
SCA	Side Channel Analysis

CHAPTER 1

INTRODUCTION

Communication, business, marketing, using smart devices are only some ways of popular devices by which people share their secrecies. Internet connections of many embedded devices such as smart phones and televisions are some of the most popular among them. Since types and methods of communications and data storage techniques are increasing dramatically, people have difficulties in keeping their information secret and communications authenticated. Therefore, security becomes important problem of governments, companies, foundations or people. Cryptography is a field providing them with sharing their data or use their rights in security. However, since there are so many different types of communication methods between people, devices or between people and devices, cryptography obtains different kinds of algorithms in order to create totally secure networks such as symmetric and asymmetric-key cryptography. For example, In Turkey [9], new identification cards are created in order that citizens can easily use them for different purposes. Authentication, electronic signature and secrecies of data are some of main properties of them provided by cryptographic algorithms. Moreover, different types of attacks are being improved and they become a threat for many secure cryptographic systems because of extensive usage area of cryptographic algorithms. Differential[8] and linear [7] cryptanalysis and side-channel analysis are some of the popular attacks improved and efficiently used nowadays. Although improving cryptanalysis methods cause high damages to cryptographic algorithms, they provide scientists to generate countermeasures or more secure algorithms. While highly secure algorithms are improved or new countermeasures increase availabilities of algorithms, updates in attacks or creations of new algorithms continue to weaken cryptographic systems. Therefore, competition between attacks and cryptographic algorithms is one of the interesting and exciting parts of cryptography. In order to understand the competition between cryptographic algorithms and attacks, algorithms and attacks can be divided into some parts. Cryptographic algorithms mainly contains three type of algorithms, namely symmetric and asymmetric-key cryptographic algorithms and keyless algorithms.

1.0.1 Asymmetric-key Cryptography

Asymmetric-key cryptography is composed of algorithms using different encryption and decryption keys. However, they are very inefficient algorithms for fast commu-

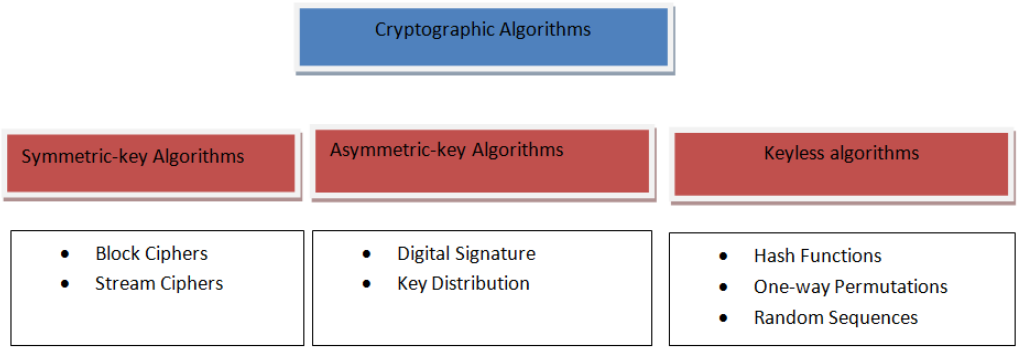

Figure 1.1: Classification of Cryptographic Algorithms [18]

nications because encryptions of plaintexts take more time than symmetric key algorithms do. Therefore, asymmetric key algorithms should be preferred in the small parts of communications such as authentication or key distribution. The difference between encryption and decryption key is the main properties of asymmetric key algorithms because if encryption key is public and decryption key is private, then everyone can encrypt plaintext and send to owner of keys but only owner can decrypt it. Similarly, keys can be distributed by encrypting with public key. Moreover, private keys can be considered as signatures of users because only users can know and use them. Because ciphertext can be decrypted by only public key, decryption with public key of users verifies that data is sent by users. Therefore, private keys can be considered as a digital signature of owners. ElGamal signature and RSA-based signature schemes are good examples of digital signature algorithms. Data authentication also benefits from asymmetric-key algorithm like digital signature because data authentication can be provided by simple benefiting difference of private and public keys. Moreover, Diffie-Helman key exchange is a good example of key distributions which based on Diffie-Helman Problem. The attacks to asymmetric key algorithms mainly focus on theory which provides security such as baby step- giant step [27] and Wiener's attack [26] which based on weakness of RSA problem for some cases.

1.0.2 Symmetric-key Algorithms

Symmetric key algorithms are algorithms that use the same key for both encryption and decryption. Unlike asymmetric key algorithms, they are commonly used in data transportation. Symmetric key algorithms are composed of simple components which provide security. Thus, they have low implementation complexities and are easily implemented. There are two types of symmetric key algorithms, stream ciphers and block ciphers. Stream ciphers can be considered as systems which generate pseudo-random digit stream. Since stream ciphers are based on recursive relation used in the design of an algorithm, mathematical properties of recursive relations determine the security levels of stream ciphers. An important point in stream cipher is the periodicity of generated digit stream. Therefore, recursive algorithms which provide the highest periodicities should be chosen. However, statistical properties of recursive relations

can be used by stream cipher attacks. Since every digit is a combination of some of previous n digit for n-bit stream cipher, the statistical attacks that use relation of each digit can successfully remove securities of stream ciphers. One of the best examples of statical attacks is attack applied on MIFARE classic cards [25] which mainly benefits from weakness of random number generator in an algorithm and simple relations between bits.

Block ciphers take plaintext and key blocks as a unit and apply different types of operations into them in order to provide security requirements such as diffusion and confusion. Main components of block ciphers are substitution boxes, permutation functions, XOR and whitening operations and round key generator. While substitution boxes are used so as to provide diffusion and confusion between bits of each block, permutation functions provide diffusion between blocks. XOR operations whose inputs are keys and intermediate values can be used in different positions in block ciphers. Whitening parts are usually applied at the beginning and at the end of algorithms. Round key generators are deterministic algorithms depending only on key values which generate different new keys for each round. Some popular examples of block ciphers are AES [10], DES [11], Serpent [12] and Present [13]. However, there are many general attacks which can be applied to any block cipher and give important information about security of block ciphers. Linear cryptanalysis [7] and differential cryptanalysis [8] are two of the most effective methods in early 1990s. Later, many block cipher algorithms were designed to be resistant against linear cryptanalysis and differential cryptanalysis.Frequencies of occurrences of output differences for constant input difference and the linear relation between plaintext and ciphertext are the main focus points of linear and differential cryptanalysis, respectively. Due to development of security of block ciphers, new cryptanalysis methods were suggested such as impossible differential cryptanalysis [14], truncated differential cryptanalysis [15], improbable differential cryptanalysis [16] and zero correlation linear cryptanalysis [17], relatively new approach. Moreover, there are different types of attacks other than linear and differential cryptanalysis such as Side-channel analysis (SCA) and fault injection. Side-channel analysis attacks are generally applied to embedded devices. Embedded devices are the most vulnerable technologies which need protection because smart phones, RFID Cards and vehicle have critical information of users. Widely usage of authentication cards and embedded devices that provide people with living simple and comfortable causes major threat for people since they can leak secret information in many different ways. SCA takes advantages of leakages which contain information about key and reveal secrecies of embedded cryptographic devices.

Since selected substitution boxes are focus points of many attacks in block ciphers, selections of substitution boxes can directly determine the security of algorithms. Therefore, there are some properties which decide the contribution of s-box into security of algorithms such as nonlinearity, differential uniformity and algebraic degree. Classification of s-boxes in terms of these properties simplify choices of s-boxes for designer because there are many different block ciphers that use s-boxes in block ciphers. Many of them are defined on F_2^n , $n \in 3, 4, 5, 6, 7, 8$. However, only s-boxes on F_2^4 are successfully classified and few good s-boxes are known and used in algorithms for large n values. Therefore, we think that classification of s-boxes for $n = 5$ helps us to understand the behaviour of all s-boxes and to choose right s-boxes for algorithms.

However,the number of s-boxes on F_2^n and current technologies allow us to classify only quadratic s-boxes with current algorithms. Therefore, in this thesis, we mainly focus on the influence of s-box choices on different methods. Also, we explain how to classify 5×5 bijective and quadratic s-boxes with efficient algorithms. Moreover, we share experimental results of some SCA attacks into AES-128 algorithms in order to show the effectiveness of SCA on secure algorithms.

1.0.3 About the Thesis

In this thesis, there are three chapters other than Introduction, namely Side Channel Analysis, S-boxes Classification and Conclusion.

The following chapter is about side channel analysis, one of the popular topics in implementation attacks. SCA attacks can be applied into algorithms more efficiently with configuration of equipments and some parameters of algorithms. Therefore, we firstly explain these parameters and discuss the relation among them. Secondly, since Differential power analysis (DPA) and Correlation power analysis (CPA) attacks are the mostly used attacks into embedded devices in SCA, preparation and recovery part of DPA and CPA attacks are explained. Lastly, we share experimental results of SCA on AES-128 with suitable equipments and try to explain relations of Signal to Noise Ratio (SNR) and the efficiency of attacks.

In the next chapter, the classification of 5×5 quadratic s-boxes are explained. Firstly, s-box generation method is described and classification algorithm is explained. 5×5 quadratic s-boxes comprise of 75 different classes. Each of classes properties are listed and compared with each other. 74^{th} and 75^{th} classes have good non-linearity and linearity. With contribution of new s-box construction method, s-boxes classes with good characteristics are found and analysed in this chapter.

In the last chapter, I give brief information about all thesis and future works.

CHAPTER 2

SIDE CHANNEL ANALYSIS

Embedded devices and smart cards are frequently used tools in transportation, shopping and identification. Also, they are one of the main components of modern cryptosystem because of usage of cryptographic algorithms. With commonly used cryptographic devices, there are much more interests to embedded cryptographic algorithms in last decades and different perspectives provide different methods with recovering key values. Particularly, side channel analysis emerged with the article[5] provides attackers with attacking algorithms from different point of views. After usage of

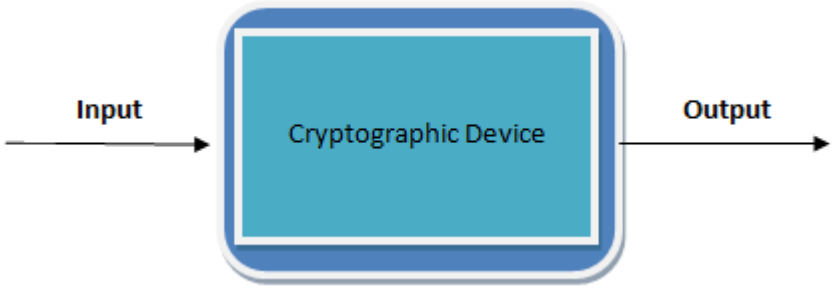

Figure 2.1: Cryptographic Device

side channel analysis on cryptographic devices, different kinds of attacks are created. While some of these attacks only listen the behaviour of devices,the others force devices to give faults and to behave unexpectedly. Therefore, attacks on cryptographic devices can be divided into two groups in terms of methods, passive and active attacks. Passive attacks only observe behaviours of cryptographic devices. However, active attacks aim to manipulate inputs and environments of devices in order to make device behave abnormally. Moreover, attacks can be classified in terms of interfaces as well as behaviours. In invasive attacks, attackers can try all different ways in order to discover secret information. Depackaging or changing functionalities of devices are examples used in invasive attacks. If attackers only try to observe data flows in devices, then the attack is called as passive invasive attacks [6]. Other than invasive attacks, there are semi-invasive attacks which only access some part of devices such as memory cells. Unlike invasive attacks, they aim to cause fault in devices by using x-ray or electromagnetic field. Lastly, non-invasive attacks only follow environmental effects of devices without causing any changes on devices. Invasive, semi-invasive and

non-invasive attacks are distinguished as passive and active. Side channel analysis is an attacks benefiting from side channels of devices such as time, power and temperate. it is considered as non-invasive and passive attacks.

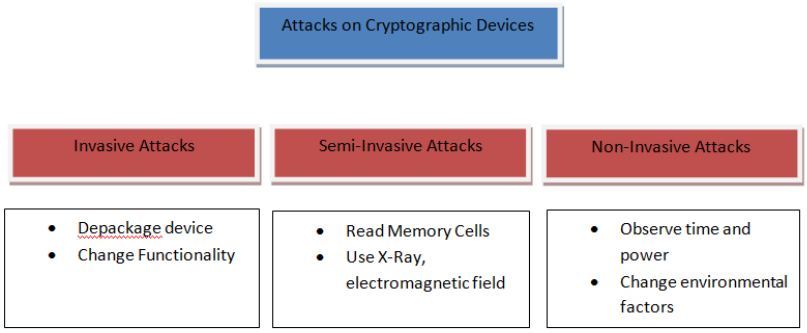

Figure 2.2: Classification of Attacks

SCA is one of the most popular analysis for embedded devices because it is non-invasive analysis, easy to apply into devices and it does not need expensive equipments [6]. Therefore, different types of attacks and new countermeasures have been improved increasingly in SCA such as power analysis attacks and time attacks. Time attacks are interested in spent time changes of embedded devices in terms of inputs or outputs. Different data can cause different operations to become active or passive. Therefore, spent time of algorithms provide us with guessing about data used by algorithms. Power consumptions of embedded devices are classified in terms of data and operations because operation and data are two factors that affect power consumption. Fundamental reason of power consumption and data relation is CMOS technologies because embedded devices are created as concatenations of many CMOS circuits. CMOS inverter is a good example in order to understand instantaneous power consumption changes. In the figure 2.3, IN value changes the transistors $M1$ and $M2$. OUT value is connected to V_{DD} or ground with respect to transistors. However, changes of IN values cause instantaneous changes in power consumptions which enable attackers to understand whether IN value changes or not. Assume that we have 8-bit register initially set. If we write 256 different values into register and measure instantaneous power consumption, we can see that the number of changing bits in the value is related to instantaneous power consumption. If a register is reset initially, the number of 1's in a register is directly proportional to power consumption because of CMOS technologies.

Since embedded devices are created by using many CMOS circuits, changes of data can cause leakages in different CMOS circuits. Therefore, data-dependent models are needed to be created to understand the behaviour of power consumption P_{DATA}. Hamming weight and Hamming distance are two of the most popular models used by power analysis attacks. Hamming weight measures number of bits in the intermediate value and can give information about how much power consumption changes. However, power consumption changes are not directly related to the number of bits for some devices but the number of changes in bits. Therefore, hamming distance can be preferred if how data changes is known by attackers because hamming distance measures the

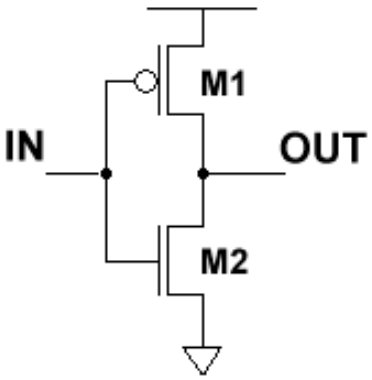

Figure 2.3: CMOS Inverter

number of changes in bits after operation is applied into data. Number of register bits used by model is as important as type of models because more the number of bits gives us a chance to have more information in power consumption.

In this chapter, it is planned to introduce side channel analysis and explain factors which change power consumption. Also, we plan to introduce two power analysis attacks, namely differential power analysis and correlation power analysis with different intermediate models. Finally, experimental results of power analysis attacks into AES-128 are discussed in terms of intermediate models, SNR and attack efficiencies.

2.1 Notations

All notations used by explaining SCA attack and countermeasures are defined in this section.

Each value of power consumption traces will be shown as $T(P, K, i, j)$ where P,K are a i^{th} words of plaintext and a key used by algorithm generating power consumption, the index of power traces is j.

$M(P, K_{guess}, i)$ is model used by SCA attacks in order to generate intermediate values where P is i^{th} word of plaintext and K_{guess}.

Substitution boxes with input P are shown as $S(P)$.

SNR value is a ratio between the signal and the noise component of a measurement. SNR value gives us information about a leakage in power consumption. [6]

$$SNR = \frac{P_{data}}{P_{noise}} \tag{2.1}$$

where P_{data} and P_{noise} is power value consumed by effects of data and power consumption of device because of environmental reasons and unused part of data , respectively.

Hamming weight of X is the number of 1 in bit representation of X.

$$HW(X) = \#\{i|(X/(2^i)) \tag{2.2}$$

Hamming distance of X,Y is the number of different bits in bit representation of X and Y.

$$HD(X,Y) = HW(X \oplus Y) \tag{2.3}$$

LSB of X is the model that returns the least significant bit value of X.

$$LSB(X) = X \tag{2.4}$$

In the following sections, word can be considered as w-bit part of data

2.2 Power-Monitoring Attacks

Considering all kind of side channel attacks, power analysis is the most popular and useful method in order to use for recovering the key. However, understanding the behaviour of power consumption is a really hard problem because there are so many dependencies of power traces and measurements of power consumption are so sensitive because the measurements of instantaneous power consumption depend on many different factors such as design, environments, oscilloscope and implementation of algorithm. Therefore, it is planned to explain factors which affect the measurement of power consumption.

Sampling Rate:

Oscilloscopes measure and save analog signals periodically. Sampling rate is equal to the number of points measured by oscilloscopes in a second. More sampling rate provides less information loss through sampling process and more sensitive digital data. Thus, it is directly proportional to efficiency of attacks but runtime of attack and space complexity of attacks is also directly proportional to sampling rate. Sampling rate should be decided in terms of performance of computer on which attack algorithm runs and efficiency of attacks.

Resolution:

Since power consumption is converted from analog to digital values, power consumption value should be reduced from all possible values to some intervals. Therefore, resolution value should be decided before measuring power consumption. Similar to sampling rate, more resolution values cause algorithm to take more time and to use more memory space. However, high resolution values provide more accurate results at the end of attacks.

Power consumption tables containing instantaneous power consumption is called as power traces. Power traces contain power consumptions which are affected by data and operations in devices and other factors. Therefore, total power consumption should be classified in order to understand the behaviour of power consumptions.

$$P_{TOTAL} = P_{DATA} + P_{OP} + P_{NOISE} + P_{CONST} \tag{2.5}$$

[6]

P_{CONST} shows constant power value which is consumed in every situation. Since P_{CONST} occurs in every situation, it is assumed that it does not affect dramatically the results of power analysis attacks. However, P_{DATA} and P_{OP} are important data and operation dependent power values, respectively. P_{NOISE} is a critical part of power values because high P_{NOISE} value can cause attacks to fail. Therefore, factors which increase P_{NOISE} should be removed or minimized. Removing P_{NOISE} and P_{CONST} obtain attackers for a clean attacks.

Assume that we generate power traces with same operations and data. Since P_{CONST} is only constant power consumption, $\mu(P_{DATA}), \mu(P_{OP}), \mu(P_{NOISE})$ and $VAR(P_{CONST})$ are zero. Since same operation and same data is used in order to generate power traces, we expect that $VAR(P_{DATA})$ and $VAR(P_{OP})$ are zero[6]. Therefore,

$$VAR(P_{TOTAL}) = VAR(P_{NOISE}). \tag{2.6}$$

It is easily understood from 2.6 that P_{NOISE} is normally distributed because P_{TOTAL} is normally distributed. In order to remove effects of P_{NOISE} from power traces, the mean of power traces generated with same data and same operations can be used in attacks. Removing P_{CONST} is an easier procedure than removing P_{NOISE}. After generating all power traces for all data, it is enough to remove $\mu(P_{DATA}) = \mu(P_{TOTAL})$ because P_{CONST} is only a constant value in power traces. [6]

After removing the effects of P_{CONST} and P_{NOISE}, remaining part can be used efficiently to understand the relation of data and power traces. Since there are n+1 different values for hamming distance of n-bit register, power distribution is a combination of n+1 normal distribution with different mean values. Therefore, power consumption value is directly proportional to hamming distance value of data. If registers always have the same data before the result of operation is written into register, hamming weight can be used instead of hamming distance.

We can understand behaviour of P_{OP} for different operations. Power consumption of different operations shows that P_{OP} and P_{DATA} are almost independent from each other. The only relation between them is that P_{DATA} values depend on the type of operation but behaviours of P_{DATA} are independent from the type of operations. However, we are only interested in power traces generated with different data and same operation in this thesis.

In embedded devices, all input bits are used and all output bits are generated in parallel. Therefore, P_{DATA} contains information about all bits. However, since SCA attacks are based on divide and conquer techniques it is not possible to benefit from all of P_{DATA}. Therefore, we can classify them as exploitable and switch noise. [6]

$$P_{DATA} + P_{OP} = P_{EXP} + P_{SW.NOISE}[6] \tag{2.7}$$

P_{EXP} is a power consumption value of data we exactly are interested in. $P_{SW.NOISE}$ is a power value generated by the the remaining parts of data. However, $P_{SW.NOISE}$ values cause noise in power traces and affect the efficiencies of attacks. SNR value is very helpful to understand relations between data length and efficiencies of attacks. However, there are disadvantages of large data such as complexity of attack. Data length, number of power traces and time efficiencies are parameters which effects each other. If length of data used by algorithm is increased in models, then SNR value also increases and the attack can be applied into algorithm efficiently. However, candidates of keys are increased and time complexity of attack algorithm converges to time complexity of random search algorithm. Therefore, data length should be decided in terms of number of power traces and cryptographic algorithm. For example, LSB is the model containing smaller data length than hamming distance and hamming weight if data length is greater than 1. Hamming weight and distance models can be used with different data length.

However, only detecting power consumption of cryptographic device during encryption or decryption is not enough to recover the secret key. We also need to know ciphertext or plaintext corresponding to power consumption and suitable mathematical model. Correlation and difference of power consumption tables are the two of fundamental techniques used in Power analysis attacks. Therefore, we plan to explain differential power analysis [2] and correlation power analysis [3] and to use them in experiments. At the end of the section, we share practical results of attacks obtained from AES algorithm on SAKURA.

2.2.1 Differential Power Analysis

Many cryptographic algorithms are used in embedded devices. Unlike operating systems, dedicated embedded devices can reflect behaviours with environmental factors. Therefore, their behaviours during encryption/decryption process consists of many information about the data used by algorithms. Since power consumptions of devices are affected by instructions done by processor and data used by algorithm, we have a chance to predict one of the data used by processors. However, it is needed to know the operation, the data used by the operation and the power consumption corresponding to data. The remaining part is to guess parts of data and reveal all key value.

Differential power analysis mainly focus on difference of means of power consumptions. Hamming weight and Hamming distance are models we can use when applying to the DPA attacks. Also, attackers can use the knowledge about implementation of algorithm in order to understand which operation causes leakages because where exactly interested instructions are applied is a very helpful information for them in order to decide the model and to decrease the time complexity of attacks.

Another thing in order to apply differential power analysis is power consumption traces. The number of plaintexts and power consumption pairs are directly proportional to success rate of DPA Attacks. Moreover, random distribution of plaintext can provide attacker with more accurate results because randomness in plain text obtain power consumption of plain text value to behave as random distribution and probabil-

ity of each candidate key value becomes independent from plaintext value.

DPA attack can be applied into plaintexts and ciphertexts. Therefore, there are two choices for attacker, the first or the last part of algorithm. WLOG, we assume that we apply the attack into algorithm from the beginning and use the first part of algorithm and plaintext in remaining part of thesis.

DPA attack is composed of two main steps, preparation and attacks.

Preperation Part:

Let T be the number of power traces, and k, n the size of key and plaintext, respectively. In this part, algorithm is repeated for random plaintexts and the same key and power consumption is measured by oscilloscope with t samples. Since oscilloscope is used in preparation parts, sampling rate and resolution values are determined.

As a result, we collect data during encryption and generate power consumption tables for each random plaintext. Power values are easily achieved by measuring the voltage difference between end points of $1\ \Omega$ resistor.

Attacks:

It is enough to compute S-box value of i^{th} word of plain text and XOR with candidate K_i value. Since there are 2^w candidate key value and T plaintexts, $2^w * T$ intermediate values are computed for each i value. Totally, $2^w * T * \frac{k}{w}$ is the number of s-box and XOR computation in order to find all key values.

Let the first guess of key be g such that $0 \leq g \leq 2^w - 1$. Then, we easily apply S-box into each i^{th} plain text word and XOR with g.

$$C_i = S(P_i) \oplus g.$$

For each g, we have hypothetical results C_i whose number is equal to T, the number of power traces.

The attack is based on relation of power consumption of algorithm and the hypothetical results $C_i's$. LSB, hamming weight and hamming distance are fundamental and general models which can be applied into many algorithms in order to generate intermediate values. Since we do not know when exactly these operations are done by the processors, the hypothetical results are compared with each sample of the power consumptions.

In DPA attack, power consumption tables are divided into two parts in terms of model outputs. Means of two groups are subtracted from each other and the result array is calculated. If we have differences of two random trace groups for wrong key guesses, it is expected to have random results because wrong keys cause model to generate wrong intermediate values and power traces are divided into two groups randomly. The number and quality of traces and selection of model are important factors which affect randomness of results for wrong keys. Because of data dependency of power consumption, it is expected that models divide traces into two groups, high and low

power trace groups for right key guess. Absolute value of difference of groups have a peak value where the operation is exactly applied. Since all other results are random values because of wrong key guess, maximum value of all peak values is corresponding to right key with highest probability.

$$Result(j, K_g) = |\frac{\sum_{i=i}^{m} T_{(P,K,i,j)} \times M_{(P,K_g,C,i)}}{\sum_{i=i}^{m} M_{(P,C,i)}} - \frac{\sum_{i=i}^{m} T_{(P,K_g,i,j)} \times (1 - M_{(P,K_g,C,i)})}{\sum_{i=i}^{m} (1 - M_{(P,K_g,C,i)})}| \tag{2.8}$$

where K_g is K_{guess}

As a result, maximum value of $Result(j, K_{guess})$ is calculated and saved as score of DPA attacks for candidate K_{guess}. For each K_{guess}, score values are computed and maximum score value is considered as real key value. However, the number of power consumption should be enough in order to find maximum score for right key. Moreover, efficiencies of the models used by DPA attack can be understood by comparison of the number of power consumption needed in order to guess the correct key with a certain high probability.

2.2.2 Correlation Power Analysis

In DPA, we can only separate traces in two groups and try to benefit from relation of traces and data. However, power consumption consists of more information than we have in DPA. Therefore, we need to use different technique which benefits from relation of traces and data. Pearson correlation is one of the best ways in order to understand linear relation between two data vectors. Although the relation between traces and data is not linear, analysing linear relation gives higher information about key values than DPA. Therefore, CPA attacks controls correlation value of intermediate results C_i and power consumption in order to understand how they are related to each other.

Only difference between DPA and CPA attacks is that $Result(j, K_{guess})$ is computed as

$$Result(j, K_{guess}) = \frac{m \times \sum_{i=i}^{m} T_{(P,K,i,j)} \times M_{(P,K_{guess},C,i)} - \sum_{i=i}^{m} T_{(P,K,i,j)} \times \sum_{i=i}^{m} M_{(P,K_{guess},C,i)}}{\sqrt{\sigma_{T_{(P,K,i,j)}}} - \sqrt{\sigma_{M_{(P,K_{guess},i,j)}}}} \tag{2.9}$$

where

$$\sigma_T = \sum_{i=i}^{m} T^2 - (\sum_{i=i}^{m} T)^2$$

Correlation formula slightly increases complexity of the attack but it gives more accurate results even with small number of power traces. Therefore, CPA attack can be preferred if there is a powerful platform for CPA algorithm.

2.3 Contribution

In this section, we plan to share experimental results of implementations of attacks described above. Main focus of the experiment is to exemplify the relation of P_{NOISE} and P_{EXP}. Therefore, correlation values of right key guesses are compared with each other for different number of bits used by attacks. In order to generate power traces efficiently we use well-known and recommended board, SASEBO SAKURA-G board [4]. Thus, following experiments are applied on power traces of AES-128 algorithm obtained by SASEBO board.

SASEBO Board:

SASEBO board is a very useful board in order to run embedded algorithm and measure power consumption. Therefore, we plan to apply our experiments on AES-128 algorithm with SASEBO SAKURA-G device. Since the board is designed in order to be used for SCA experiments, it provides very comfortable platform for users to generate power traces efficiently. There are USB and RS-232 connection with computers and GUI for encryption tests. Since source code of GUI is provided on internet, with some modification many encryptions and power consumption measurements can be completed in short time.

Oscilloscope:

Digital Oscilloscope is used for sampling instantaneous power consumption values. In our experiments, Keysight 3000A X-series is used. It has Ethernet interface which provides connection to personal computer. Two probes of oscilloscope is connected to beginning and end of 1Ω register on power line.

Personal Computer

Personal Computer (PC) has connection with SASEBO Board and Digital Oscilloscope. Firstly, PC send a plaintext and key values to SASEBO Board and trigger message to SASEBO Board in order to start encryption. Secondly, it starts to listen data that come from oscilloscope. This procedure is repeated by PC until enough number of power traces is collected.

Different kinds of power traces can be generated because of different equipments used or probe positions. Design of board and environmental effects and probe connections are some of critical factors affecting power traces graphics [6].

In our experiments, we aim to see effects of the number of power consumptions, SNR and models (LSB, Hamming weight, Hamming distance) into efficiencies of DPA and CPA attacks. Only LSB is used for DPA attacks and the attack is repeated for different T values. CPA attack is applied into 10000 power traces by using different number of

bits in hamming distance model.

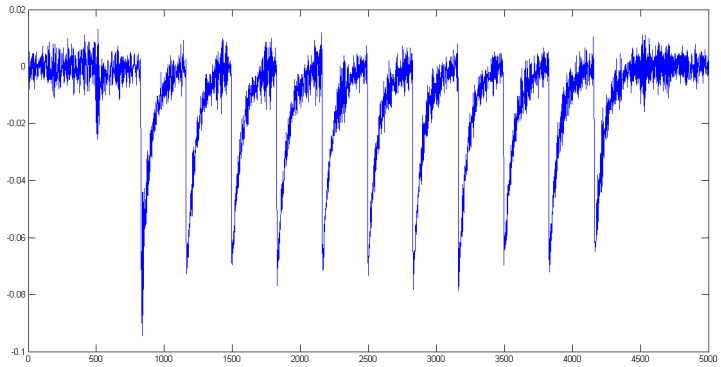

Figure 2.4: Power consumption Trace of AES-128

It is easily seen that DPA attack takes a few seconds to return the result if it runs on double core computer with 2.53 Ghz processor and 4 GB ram. However, because of many correlation computation, CPA attack takes some minutes in same platform when applied to 10000 power traces. Since both of DPA and CPA attacks are time efficient algorithms, it becomes more efficient to focus on efficiencies of attacks in terms of SNR and the number of power traces.

Until 1000 traces, it is impossible to separate right key from the others but between 1000 and 4000 traces, DPA result of right key value starts to appear in highest three values. After 4000 traces, it is clearly understood that right key value is the highest value calculated DPA attack.

Number of traces and number of bits used by models are important parameters which decide efficiencies of attack algorithm. Since efficiency of algorithm can be understood from correlation value of right key guess. SNR value of 8-bit scenario is expected to be higher than SNR value of 1-bit scenario because more bit changes are included by model, stronger relation between model and power traces is obtained [6]. In practical results in figure 2.5, it is easily seen that until 2000 traces, results of right key guess changes randomly. Therefore, effects of SNR values are discussed when power traces are between 2000 and 10000. Right key starts to have the highest value when the number of bits used by hamming distance model is greater than 6. For small number of bits used by the model, result of algorithm for right key changes between 0.06 and 0.04 which is close to random correlation values.

Choice of model is also an important factor in terms of efficiency. For example, in our experiments we see 1000 decrease in the number of traces when we use hamming distance model instead of hamming weight model.

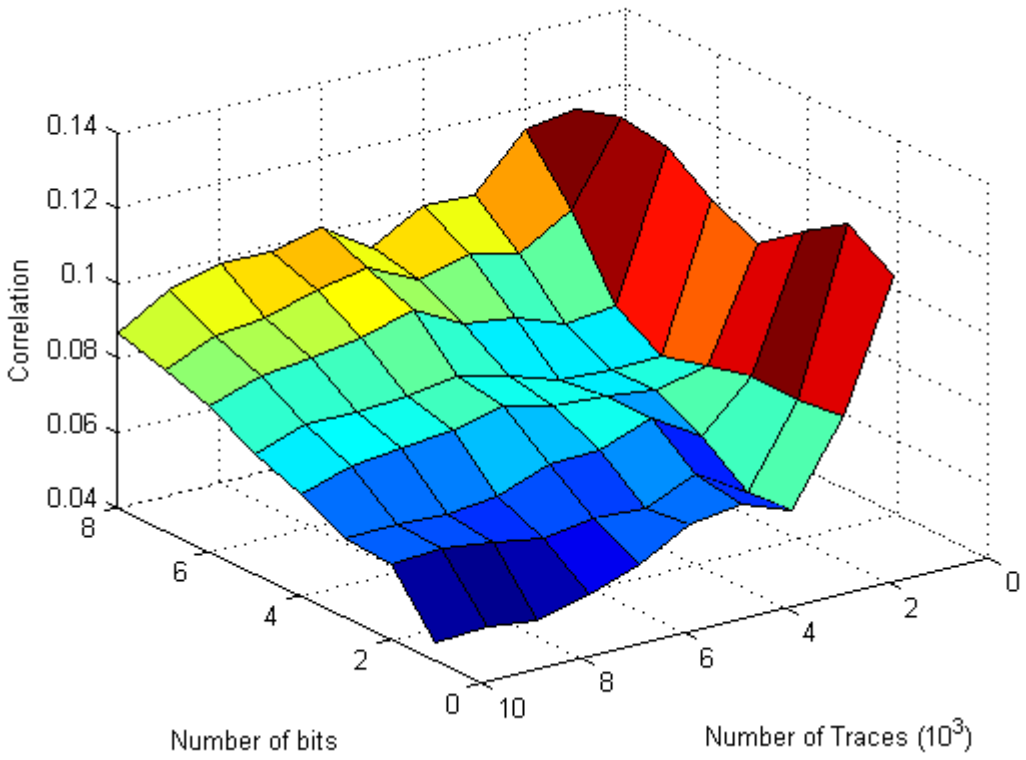

Figure 2.5: Effects of Number of bits used by Model in DPA-Hamming Distance Attacks

2.4 Countermeasures

Although DPA and CPA are efficient power analysis attacks, there are fundamental and efficient countermeasures against them. All side channel analyses need enough power traces and plaintext (ciphertext) pairs. Therefore, preventing attackers from collecting enough traces is the easiest solution by changing symmetric keys more frequently. However, it is not a practical solution because cryptographic devices should also be suitable for long term usage. Another approach is to construct new hardware or software methods which decrease leakages and prevent attackers from revealing the key. Since the relation between power traces and intermediate value generated by attacker is a main reason which causes information leakage, new methods should randomize intermediate value or power consumption [1] for all models. Secret sharing is one of these methods which we are interested in this section. Secret sharing prevents appearances of intermediate values in run time of algorithm and power consumption table is generated independently from intermediate values used by SCA attacks. Since each round in block cipher consists of substitution function, sharing can be provided by changing inputs and methods without changing function. Firstly, each input variable is divided into at least $d + 1$ shares for d^{th} order sharing and each value is used in computations independently. Therefore, power consumption is not affected as much

as direct computation.

Assume that we have a function

$$F(X, Y, Z) = X \times Y \oplus Z$$

and we would like to apply sharing into a function F [1]. Let $X = (X_1 \oplus X_2)$, $Y = (Y_1 \oplus Y_2)$ and $Z = (Z_1 \oplus Z_2)$. We need to change function F in order to generate same result with input shares. However, critical point in construction of F is not to appear X, Y and Z because appearance of inputs in computation causes leakage with power consumption. Then new function F' is defined as,

$$F'(X_1, X_2, Y_1, Y_2, Z_1, Z_2) = X_1Y_1 \oplus (X_2Y_2 \oplus (X_1Y_2 \oplus X_2Y_1 \oplus Z_1)) \oplus Z_2$$

If we sum firstly $X_1Y_1 \oplus X_1Y_2 = X_1Y$, then Y will appear and cause information leakages about Y value. Moreover, d^{th} order sharing methods are applied into functions by distinguishing inputs into $d + 1$ shares.

Another important point is the multiplication number in function F. Multiplication number in function F increases the number of summations and multiplications dramatically. Therefore, quadratic functions are a good choices in order to avoid disadvantages of multiplication. If cubic function was selected then eight multiplication would be needed in order to compute the result for one multiplication in cubic function.

CHAPTER 3

S-BOXES CLASSIFICATION

If a new cryptographic algorithm is created, the first thing is to select good s-box. However, "what is the good S-box ?" is one of the hard questions to answer in cryptography for $5, 6, 7$ and 8 dimensional field. Although it is hard to select the most effective S-box for high dimension field, some characteristics of S-boxes are very helpful to classify and determine efficiencies of s-boxes such as linearity, uniformity, algebraic degree and gate complexity. For 4×4 and 3×3 S-boxes, all classes are decided in terms of affine equivalence and charateristics of all class are listed in [1]. Thanks to classification of 4x4 S-boxes, S-boxes with desired linearity or algebraic degree can be selected and used in the algorithm. In this way, security can be obtained againts the attack which benefits from properties of S-box up to some extend. Although good selection of S-boxes increases security through mathematical analysis, SCA can cause a major threat for security algorithms because SCA is one of the most popular field which make use of not only mathematical properties of the algorithm but also implementation of the algorithm. Therefore, cryptographic algorithm should be implemented in such a way that information leakage should be minimized. However, minimizing leakages or randomizing intermediate values is not easy process and is usually costly operations. Beside countermeasures, selecting component of algorithm secure against power analysis attacks becomes so important with different kinds of attacks.

In this chapter, definitions of affine equivalence and algorithm *Finding Affine Equivalence* is described. Then, construction method of 5×5 permutations and main algorithm is explained. In the last part of the chapter, properties of all quadratic classes over F_2^5 are listed and compared to each other. Moreover, invertible S-boxes is focused and classified in this chapter and they are called as "permutation".

The study in this chapter are the joint work with Dusan BOZILOV.

3.1 Preliminaries

The best classification of permutations over F_2^n is provided with affine equivalence because it preserves some important properties of s-boxes such as linearity, differential uniformity and algebraic degree. In $n \times n$ permutation, the classification of permutations is defined with affine equivalence. Definition of affine equivalence is as follows

Definition 3.1. Let S_1 and S_2 be permutations over F_2^n. S_1 and S_2 are called as affine equivalent if there exist A and B such that $S_1 = A \times (S_2 \oplus a) \times B^{-1} \oplus b$ where A and B are $n \times n$ invertible matrices over F_2^n.

Affine equivalence of two s-boxes can be determined by applying random search algorithm to matrices A and B. However, the complexity of random search algorithm is $O(2^{2n^2})$. Complexity of classification of all s-boxes is $O(n! \times 2^{2n^2})$. With this complexity, random search algorithm is not expected to be succeed successfully in current technology. Therefore, more efficient algorithm than random search algorithm is needed for the classification. The algorithm given by De Canniere [22] is the most efficient algorithm with complexity $O(2^{3n})$ successfully classifying s-boxes for $n = 4$ and 5.

3.1.1 Finding Linear Representative

The algorithm explained in this chapter is used to find linear representative of input s-box. Important point in the algorithm is to find a suitable A and B. Unlike naive approach, the algorithm benefits from linearity of A and B. Firstly, some points of A and B are guessed and linearities are controlled. If linearity of at least one matrix is not preserved , then previous guesses are changed and algorithm continues until generation of A and B.

Definition 3.2. Linear representative of S-box S, R_S is the lexicographically smallest S-box in the linear equivalence class containing S.

In order to better understand the algorithm, some sets are defined.

Sets D_a and D_b:These sets contain points which linear mappings of A and B are defined.

Sets C_a and C_b:Since A and B are dependent to each other, elements in D_a and D_b are controlled whether they are contradict to each other. C_A and C_B consist of elements of D_A and D_B which preserve followings,

$$C_A \subset D_A, C_B \subset D_B \tag{3.1}$$

$$C_B = B^{-1} \circ S \circ A(C_A) \tag{3.2}$$

$$C_A = A^{-1} \circ S^{-1} \circ B(C_B) \tag{3.3}$$

By 3.1 and 3.3,

$$C_A = A^{-1}(A(D_A) \cap S^{-1} \circ B(C_B)) \tag{3.4}$$

By 3.1 and 3.2

$$C_B = B^{-1}(B(D_B) \cap S \circ A(C_A)) \tag{3.5}$$

Sets N_A and N_B:These sets consist of elements which linear mappings of A and B are defined but linearities of A and B are not checked yet.

$$N_A = D_A \setminus C_A \tag{3.6}$$

$$N_B = D_B \setminus C_B \tag{3.7}$$

Since $A(0) = 0$ and $B(0) = 0$, D_A and D_B are initially equal to set $\{0\}$. By 3.6 and 3.7, initial values of C_A and C_B are set.

If $S(0) = 0$, then $C_A = C_B = D_A = D_B$. Clearly, by definition, N_A and N_B are empty. Therefore, algorithm start guessing some points of A.

Otherwise, C_A and C_B are empty sets and N_A and N_B consists of only 0 as element.

Data: S
Result: Result : Linear Representative
initialization;
while $N_A \neq \varnothing$ **do**
    pick $x = min(\overline{C_A}) = min(N_A)$
    pick $y = min(\overline{D_B}) = |D_B|$
    assign $R'_s(x) = y$ and $D_B = D_B \oplus y$
    Update all sets
    while $N_A = \varnothing$ *and* $N_B \neq \varnothing$ **do**
        pick $x = min(\overline{C_A}) = min(\overline{D_A}) = |D_A|$
        pick $y = min(\overline{C_B}) = min(N_B)$
        assign $R'_s(x) = y$ and $D_A = D_A \oplus x$
        update all sets
    end
end

Algorithm 1: Finding Linear Equivalence Algorithm

In order to find affine equivalence, finding linear equivalence algorithm should be applied for all a, b.

3.2 Contribution

3.2.1 New representation of permutations

Permutations are generally represented as a vectorial boolean functions. However, new representation enabling us to efficiently generate all quadratic permutations is required. Therefore, matrix representation of algebraic normal form of permutations is preferred because it gives a chance to generate only quadratic boolean functions. In this section, it is explained how to represent algebraic normal form as a matrix. Also, new matrix is created in order to convert representation of boolean functions from algebraic normal form to standard form and there is a simple algorithm at the end of section that is used in construction of permutations from boolean functions.

Definition 3.3. Assume that S is a permutation defined over F_2^n. The $n \times 2^n$ matrix generated by vectorial representations of coefficients of terms in algebraic normal form of S. It is called as S_{ANF} [22]

Example:
Let output of S be defined as

$$y_1 = x_1 \oplus x_3,\ y_2 = x_1 \oplus x_2x_3,\ y_3 = x_2 \oplus x_1x_3$$

Then, $S_{ANF} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ where columns are constant term and coefficients of $\{x_1, x_2, x_1x_2, x_3, x_1x_3, x_2x_3, x_1x_2x_3\}$

Finding linear equivalence algorithm requires the standart representation of each s-box. Therefore, it is needed to create another matrix 3.4 which turn S_{ANF} into standart representation.

Definition 3.4. I_{Bool} is a $2^n \times 2^n$ matrix whose columns consist of vectorial representation of terms result in algebraic normal form. In other words, i^{th} column consists of values of terms in algebraic normal form when x is equal to i

In our structure, standard representation S_{TT} is camputed as $S_{ANF} \times I_{Bool}$.

Example: Let $S_{ANF} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$ and

$$I_{BOOL} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Then, $S_{TT} = S_{ANF} \times I_{BOOL}$ is equal to $\begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$ and

$S = (76543210)$.

3.2.2 Construction of S_{ANF} for 5 $\times$ 5 quadratic permutations

There are only 302 equivalence class in F_2^4. Only 6 of them are quadratic classes and the remaining 295 are cubic classes. For $n = 5$, there are so many equivalence classes

that it is almost impossible to list all classes and compare to each other in terms of characteristics of s-boxes. Some extensions of *Finding linear equivalence algorithm [22]* diminish the number of candidate s-boxes and make the algorithm possible to complete classification of quadratic classes. Therefore, only quadratic s-boxes are created and classified in this chapter. Moreover, properties of classes are explained and compared to each other

S_{ANF} is a 5×32 matrix for n = 5 and each column is corresponding to coefficients of terms in algebraic normal form. Since we only want to focus on quadratic terms, coefficients of terms with 3^{th} , 4^{th} and 5^{th} degree are zero. Therefore, number of columns is reduced from 32 to 15. Each column corresponds to x^1, x^2, x^3, x^4, x^5, x^6, x^8, x^{10}, x^{12}, x^{16}, x^{18}, x^{20}, x^{24} respectively. Then the structure of S_{ANF} for quadratic s-boxes is

$$\begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & c_{1,5} & c_{1,6} & c_{1,7} & c_{1,8} & c_{1,9} & c_{1,10} & c_{1,11} & c_{1,12} & c_{1,13} & c_{1,14} & c_{1,15} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} & c_{2,5} & c_{2,6} & c_{2,7} & c_{2,8} & c_{2,9} & c_{2,10} & c_{2,11} & c_{2,12} & c_{2,13} & c_{2,14} & c_{2,15} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} & c_{3,5} & c_{3,6} & c_{3,7} & c_{3,8} & c_{3,9} & c_{3,10} & c_{3,11} & c_{3,12} & c_{3,13} & c_{3,14} & c_{3,15} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} & c_{4,5} & c_{4,6} & c_{4,7} & c_{4,8} & c_{4,9} & c_{4,10} & c_{4,11} & c_{4,12} & c_{4,13} & c_{4,14} & c_{4,15} \\ c_{5,1} & c_{5,2} & c_{5,3} & c_{5,4} & c_{5,5} & c_{5,6} & c_{5,7} & c_{5,8} & c_{5,9} & c_{5,10} & c_{5,11} & c_{5,12} & c_{5,13} & c_{5,14} & c_{5,15} \end{pmatrix}$$

Lemma 3.1. *[23] Let S be an n-bit permutation. Then S is an affine equivalent to another permutation S' with $S'(x) = x for x \in \{0, 1, 2, 4, 8, ..\}$*

By Lemma 3.1, it is understood that it is enough to check s-boxes such that $S(x) = x, x \in \{0, 1, 2, ..\}$ in order to find all quadratic partitions because lemma shows us that every class consists of at least one s-box S' in lemma. Therefore, the columns $1^{st}, 2^{nd}, 4^{th}, 7^{th}$ and 11^{th} are set as $\{1, 0, 0, 0, 0\}$, $\{0, 1, 0, 0, 0\}$, $\{0, 0, 1, 0, 0\}$, $\{0, 0, 0, 1, 0\}$, $\{0, 0, 0, 0, 1\}$, respectively. This property provides reducing the number candidate rows from 2^{15} to 2^{10}.

As a result, structure of S_{ANF} used in construction of 5×5 permutations is

$$S_{ANF} = \begin{pmatrix} 1 & 0 & c_{1,3} & 0 & c_{1,5} & c_{1,6} & 0 & c_{1,8} & c_{1,9} & c_{1,10} & 0 & c_{1,12} & c_{1,13} & c_{1,14} & c_{1,15} \\ 0 & 1 & c_{2,3} & 0 & c_{2,5} & c_{2,6} & 0 & c_{2,8} & c_{2,9} & c_{2,10} & 0 & c_{2,12} & c_{2,13} & c_{2,14} & c_{2,15} \\ 0 & 0 & c_{3,3} & 1 & c_{3,5} & c_{3,6} & 0 & c_{3,8} & c_{3,9} & c_{3,10} & 0 & c_{3,12} & c_{3,13} & c_{3,14} & c_{3,15} \\ 0 & 0 & c_{4,3} & 0 & c_{4,5} & c_{4,6} & 1 & c_{4,8} & c_{4,9} & c_{4,10} & 0 & c_{4,12} & c_{4,13} & c_{4,14} & c_{4,15} \\ 0 & 0 & c_{5,3} & 0 & c_{5,5} & c_{5,6} & 0 & c_{5,8} & c_{5,9} & c_{5,10} & 1 & c_{5,12} & c_{5,13} & c_{5,14} & c_{5,15} \end{pmatrix}$$

Another reduction is provided by the balance of each row. Since each row of S_{TT} is candidate of boolean functions of S-boxes and S-boxes are permutations, it is enough to take into account only balanced rows of S_{TT}. Therefore, the balancedness are controlled at steps of generation of s-boxes. The balancedness control provides reduction of candidate rows from 2^{15} to 472. Since 5 boolean functions selected from 5 different sets are needed in order to generate quadratic permutations, total number of candidate s-boxes is at most 472^5. However, balancedness of composition of boolean functions are checked in each selection. Therefore, the number of candidate s-boxes becomes less than 2^{29}.

3.2.3 Main Algorithm

Until now, balanced boolean function sets called R_1, R_2, R_3, R_4 and R_5 are generated for each coordinate of S-boxes.

Data: R_1,R_2,R_3,R_4 and R_5
Result: Result : List of Affine Class of Quadratic 5x5 S-boxes
initialization;
while $r1 \in R_1$ **do**
  while $r2 \in R_2$ **do**
    if *NotBalanced(*$r1 + 2 * r2$*)* **then**
      Continue with new $r2$.
    end
    while $r3 \in R_3$ **do**
      if *NotBalanced(*$r1 + 2 * r2 + 4 * r3$*)* **then**
        Continue with new $r3$.
      end
      while $r4 \in R_4$ **do**
        if *NotBalanced(*$r1 + 2 * r2 + 4 * r3 + 8 * r4$*)* **then**
          Continue with new $r4$.
        end
        while $r5 \in R_5$ **do**
          if *NotBalanced(*$S = r1 + 2 * r2 + 4 * r3 + 8 * r4 + 16 * r5$*)* **then**
            Continue with new $r5$.
          end
        end
        $LR = FindAffEquivalence(S)$
        if *LR* $\notin$ *Result* **then**
          Result = Result + {LR}
        end
      end
    end
  end
end

Algorithm 2: The Main Algorithm

Balancedness is checked in each step and quadratic permutation are generated. After the generation of s-box is completed, the algorithm which find linear representative is called. Result of the algorithm is checked whether a new class is found or not. The function $FindAffEquivalence$ returns the smallest linear representative of all equivalences of input s-box in terms of lexicographical order. Since smallest s-box in lexicographical order is unique, it is very helpful to use smallest linear representative of s-boxes in order to identify classes.

3.2.4 Equivalence Classes of 5 x 5 quadratic permutations

The algorithm generates 75 quadratic 5×5 classes. In this section, all quadratic classes are analysed in terms of linearity, differential uniformity, algebraic degree and complexity.

Linearity:
Linear approximation table is the best way to understand behaviour of permutations. Since attacks are benefited from high bias values, the maximum absolute values of entries of all permutations can be compared to each other. Therefore, linearity is defined as

Definition 3.5. Linearity of S-box s is defined as $L_s = max\{L_s(u,v)|u,v \in F_2^n\}$

Linearity	number of Classes	Classes
4	2	C_{74}, C_{75}
8	20	
16	53	

Table 3.1: Linearity of Classes

5×5 permutation classes have linearity $4, 8$ and 16. Only 2 classes are close to ideal s-boxes, remaining s-boxes have linearity values, 8 and 16.

The bias values of three s-boxes are $1/8$, $1/4$ and $1/2$. Therefore, only 2 classes with linearity 4 is the best choices in order to construct algorithm secure against linear attacks.

Differential Uniformity:
Differential uniformity defines behaviour of s-box with constant input and output difference.

Definition 3.6. Differential uniformity of s-box s is defined as

$N_s = max\{N_s(\Delta X, \Delta Y)|\Delta X, \Delta Y \in F_2^n\}$

N_s	number of Classes	Classes
2	2	C_{74}, C_{75}
4	3	C_{71}, C_{72}, C_{73}
8	14	
16	33	
32	23	

Table 3.2: Differential uniformity of Classes

Unlike linearity, differential uniformity of s-box classes changes between 2 and 32. Classes with 2 linearity has 2 differential uniformity. Therefore, choice of these classes provide highly secure algorithm against linear and differential uniformity. However,

some differential attacks use the bitwise representation of difference of inputs. Therefore, selecting s-boxes in terms of both differential uniformity and undisturbed bits or differential factors can be crucial.

Other properties: Some properties of S-boxes are not preserved under affine equivalence. Therefore, selected S-box can be analysed in terms of them. One of the popular parameters used in SCA is transparency order[24]. Transparency order (T_s) of s-box s is a parameter which defines influence of s-box into correlation value of correct key. For 5×5 permutations, Value T_s changes between 0 and 5. If every coordinate of s-box S is bent function, then T_s is equal to 5.

3.2.5 Multiplicative Complexity

Implementations of S-boxes are so critical in embedded devices. Optimization of non-linear gates in S-boxes simplify to extend cryptographic algorithms with countermeasure methods such as random mask or sharing. Therefore, multiplicative complexity can be checked before selection of s-boxes.

Definition 3.7. [28] Multiplicative complexity of s-boxes is the minimum number of non-linear gates(ANDs and ORs) used when the s-box is implemented with AND,XOR,NOT and OR.

However, it is not an easy procedure to find multiplicative complexity. In [28], the problem is converted to Satisfiability (SAT) problem, one of the NP-Complete problem. Therefore, some boundary values can be found by sat solvers but for some classes, exact value of multiplicative complexity is unknown. After applying the sat solver into all quadratic permutations, the following table is created.

MC	# of Classes	Classes
1	1	C_1
2	5	C_{2-4}, C_{14}, C_{18}
3	15	$C_{5-7}, C_{12}, C_{13}, C_{15}, C_{19-21}, C_{23}, C_{25}, C_{28}, C_{30}, C_{33}, C_{41}$
4	22	$C_{8-11}, C_{16}, C_{17}, C_{22}, C_{24}, C_{26}, C_{29}, C_{31-32},$ $C_{34}, C_{37-39}, C_{42-44}, C_{50-51}, C_{54}$
5	19	$C_{27}, C_{35-36}, C_{40}, C_{45-49}, C_{52}, C_{53}, C_{55-56}, C_{58-60}, C_{68}, C_{70}, C_{72}$
6	11	$C_{57}, C_{61-67}, C_{69}, C_{71}, C_{73}$
7	1	C_{75}
8	1	C_{74}

Table 3.3: Multiplicative Complexity of 5x5 quadratic Permutation Classes

CHAPTER 4

CONCLUSION AND FUTURE WORKS

SCA is one of popular passive and non-invasive attack types which applied into cryptographic devices. Analysing and exemplifying popular attacks on AES-128 in terms of critical points of SCA algorithms is very helpful steps to understand power of SCA on embedded cryptographic devices. Therefore, DPA and CPA attacks are repeated on power traces of AES-128 generated by SASEBO SAKURA-G Board with different SNR values, number of traces. Effectiveness of algorithms is directly related to use number of bits used by model. In our example, this relation and influences of number of trace into this relation is clearly showed. In our experiments, we see that right key is correctly guessed after number of traces is greater than 4000 in DPA. Also, if the number of bits used in model is greater than 6, then correlation value of right key can be clearly distinguished from others.

S-boxes are one of the main components of symmetric key algorithms. Therefore, effects of s-boxes into algorithms in term of time and space complexity and security have been studied by different kinds of researchers. Also, because of critical responsibilities of s-boxes in cryptographic algorithms, many cryptanalysis methods are interested in properties of s-boxes and new methods are improved in order to attack symmetric key algorithms. Therefore, power analysis attacks are analysed and exemplified in order to understand effects of s-boxes in different attacks. Moreover, checking of all different s-boxes and listing them in terms of critical properties are the easiest method to prevent different attacks to be applied into cryptographic algorithms. However, size of s-boxes spaces increasing dramatically and methods more efficient than random search algorithms are improved and applied. Since classification of s-boxes in terms of cryptographic properties is the best way to understand s-boxes, classification of s-boxes becomes critical steps to understand behaviours of cryptographic algorithms.

The algorithm [22] used in classification of 4×4 bijective s-boxes is the most efficient algorithms in order to classify s-boxes. However, there are time efficiency problem in classification of 5×5 s-boxes. Although classification of all bijective 5×5 s-boxes is almost impossible with today's resources, quadratic s-box classification can provide designer to select best quadratic s-boxes for their purposes. Because of new construction method of s-boxes, 75 different s-box classes are listed successfully. In terms of linearity and differential uniformity, C_{74} and C_{75} are more secure s-box classes than others'. However, multiplicative complexities of them are 8 and 7, respectively. Therefore, selected s-boxes can cause performance problem when used in embedded

device with countermeasures. Classes $C_{71-72-73}$ are second best group which have 4-uniformity s-boxes with 8 linearity. Also, multiplicative complexity of $C_{71,73}$ and C_{72} is 6 and 5 respectively. Transparency order [24] of s-boxes is not preserved under affine equivalent class. Therefore, selected s-box can be evaluated in terms of transparency order individually.

Classification of cubic 5×5 s-boxes and quadratic 6×6 s-boxes are second goal in order to access by improving the algorithm we use. However, classification algorithm runs so inefficiently with 6×6 s-boxes because complexity of algorithm is $O(2^{3n})$ where n is dimension of s-boxes. Therefore, s-boxes construction method can be improved by checking linearity and differential uniformity before classification algorithm checks class of s-boxes.

REFERENCES

[1] B. Bilgin "Threshold Implementations: As Countermeasure Against Higher-Order Differential Power Analysis", PhD Thesis at KU Leuven, Leuven, Belgium and UTwente, Enschede, The Netherlands.

[2] P. Kocher, J. Jaffe, and B. Jun "Differential Power Analysis", Advances in Cryptology - CRYPTO'99, pp 388-397,1999

[3] E. Brier, C. Clavier, and F. Oliver "Correlation Power Analysis with a Leakage Model", Cryptographic Hardware and Embedded Systems - CHES 2004, pp 16-29,2004

[4] `http://satoh.cs.uec.ac.jp/SAKURA/`

[5] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," Advances in Cryptology: Proceedings of CRYPTO' 96, Springer-Verlag, August 1996, pp. 104–113.

[6] S. Mangard, E. Oswald, T. Popp "Power Analysis Attacks Revealing the Secrets of Smart Cards", Springer- pp 70-73,2006

[7] M. Matsui, "Linear Cryptanalysis Method for DES Cipher", Advances in Cryptology-EUROCRYPT '93 (Lecture Notes in Computer Science no. 765), Springer-Verlag, pp. 386-397, 1994.

[8] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems", Journal of Cryptology, vol. 4, no. 1, pp. 3-72, 1991.

[9] `https://ekimlikrandevu.nvi.gov.tr/Pages/homepage.aspx`

[10] P. Ing Christof, and I. Jan Pelzl. "The advanced encryption standard (AES)." Understanding Cryptography. Springer Berlin Heidelberg, 2010. 87-121.

[11] C. Don. "The Data Encryption Standard (DES) and its strength against attacks." IBM journal of research and development 38.3 (1994): 243-250.

[12] A. Ross, E. Biham, and L. Knudsen. "Serpent: A proposal for the advanced encryption standard." NIST AES Proposal 174 (1998): 1-23.

[13] B., Andrey, et al. "PRESENT: An ultra-lightweight block cipher." International Workshop on Cryptographic Hardware and Embedded Systems. Springer Berlin Heidelberg, 2007.

[14] E. Biham, A. Biryukov, A. Shamir, "Miss in the Middle Attacks on IDEA and Khufu", proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 124–138, Springer-Verlag, 1999.

[15] L. R. Knudsen “Truncated and higher order differential”, In B. Preneel, editor, Fast Software Encryption-Second International Workshop, volume 1008 of Lecture Notes in Computer Science, pages 196–211. Springer-Verlag, 1995.

[16] C: Tezcan, “The improbable differential attack: cryptanalysis of reduced round CLEFIA”, In: Gong, G., Gupta, C.K. (eds.) INDOCRYPT 2010. LNCS, vol. 6498, pp. 197–209. Springer, Heidelberg (2010)

[17] B. Andrey and M.Wang. “Zero correlation linear cryptanalysis with reduced data complexity.” Fast Software Encryption. Springer Berlin Heidelberg, 2012.

[18] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, “Handbook of Applied Cryptography”, CRC Press, 1997.

[19] C. Blondeau, K. Nyberg, “New Links between Differential and Linear Cryptanalysis”, In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 388–404. Springer, Heidelberg (2013)

[20] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui,S. Moriai, J. Nakajima, T. Tokita, “Camellia: A 128-bit block cipher suitable for multiple platforms”, In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 41–54. Springer, Heidelberg (2001)

[21] D. Hong, J. Sung, S. Hong, J: Lim, S. Lee, B. Koo,C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim,J. Kim, and S. Chee “HIGHT: A New Block Cipher Suitable for Low-Resource Device”, In Louis Goubin and Mitsuru Matsui (editors), CHES 2006, volume 4249 of Lecture Notes in Computer Science, pages 46-59. Springer, 2006.

[22] C. De Cannière “Analysis and Design of Symmetric Encryption Algorithms”, Chapter 5, PhD Thesis,2007.

[23] G.Leander and A. Poschmann “Arithmetic of Finite Fields: First International Workshop”, WAIFI 2007,Madrid,Spain, june 21-22,2007.Proceeding, chapter On the classification on 4 Bit S-Boxes,pages 159-176. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[24] E.Prouff “Fast Software Encryption: 12th International Workshop”, FSE 2005, Paris, France, February 21-23, 2005, Revised Selected Papers,2005. DPA Attacks and S-boxes, pages 424-441. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[25] G.d.K Gans, J.Hoepman and F.D.Garcia “A Practical Attack on the MIFARE Classic”, In:Proceedings of the 8th Smard Card Research and Advanced Application Workshop (CARDIS 2008). LNCS, vol.5189,pp.267-282. Springer,Heidelberg (2008).

[26] R. Pinch “Extending the Wiener attack to RSA-type cryptosystems”, Electronics Letters, Vol. 31 (1995) pp. 1736–1738

[27] J.S. Coron, D. Lefranc, G. Poupard, “ A New Baby-Step Giant-Step Algorithm and Some Applications to Cryptanalysis”, In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 47–60. Springer, Heidelberg (2005)

[28] K. Stoffelen “Optimizing s-box implementations for several criteria using sat solvers”, Fast Software

Encryption, 2016.

APPENDIX A

DPA and CPA Attack Results

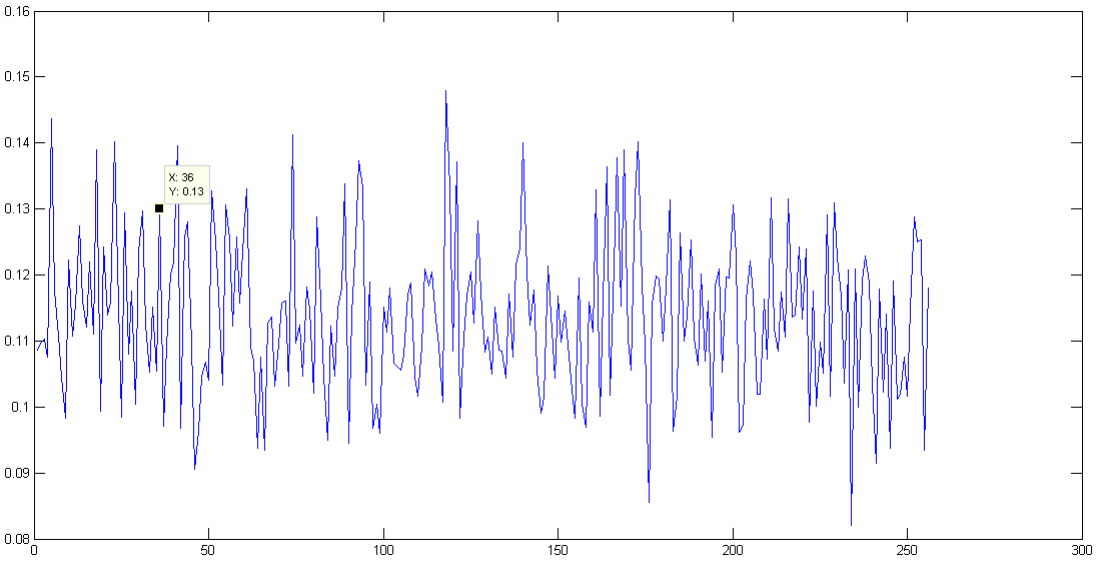

Figure A.1: DPA Attack into First Key with 1000 traces

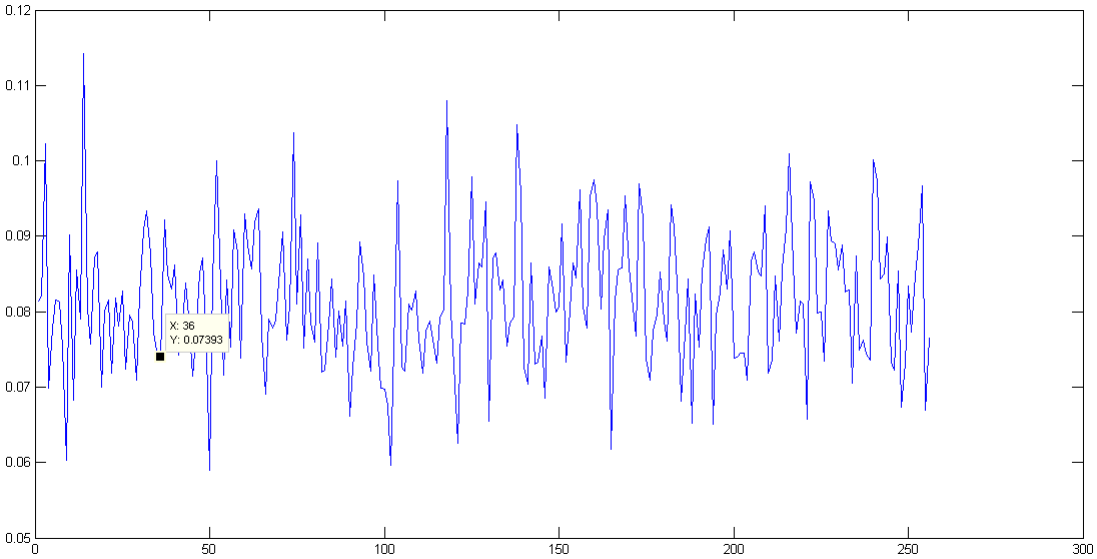

Figure A.2: DPA Attack into First Key with 2000 traces

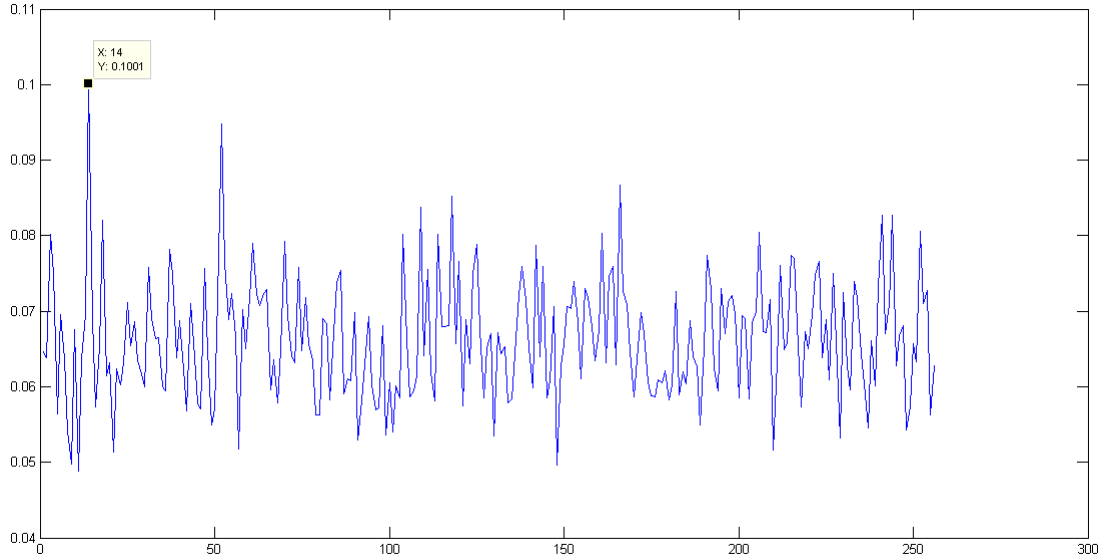

Figure A.3: DPA Attack into First Key with 3000 traces

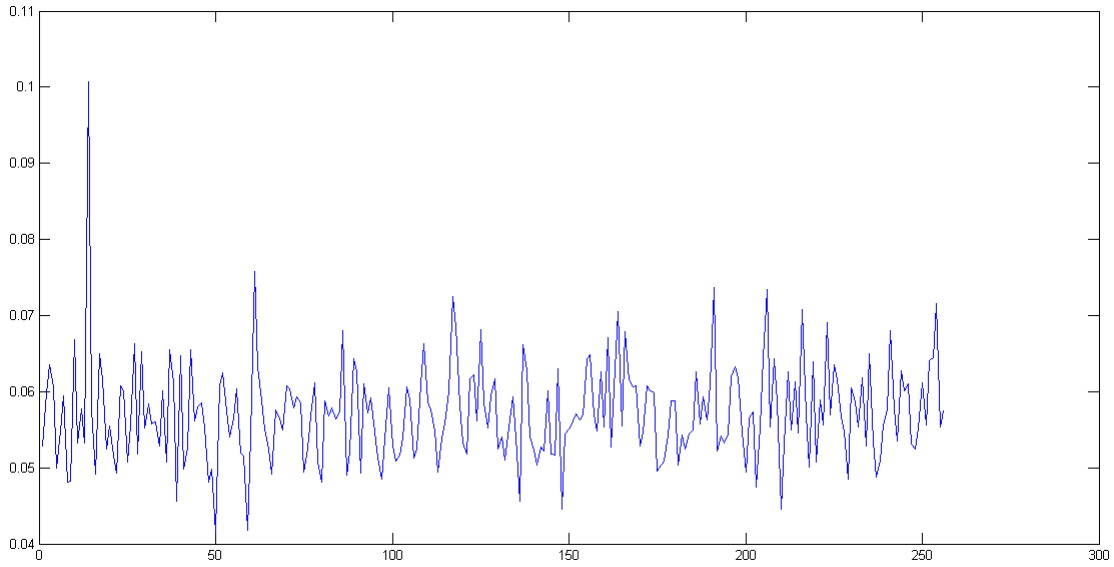

Figure A.4: DPA Attack into First Key with 4000 traces

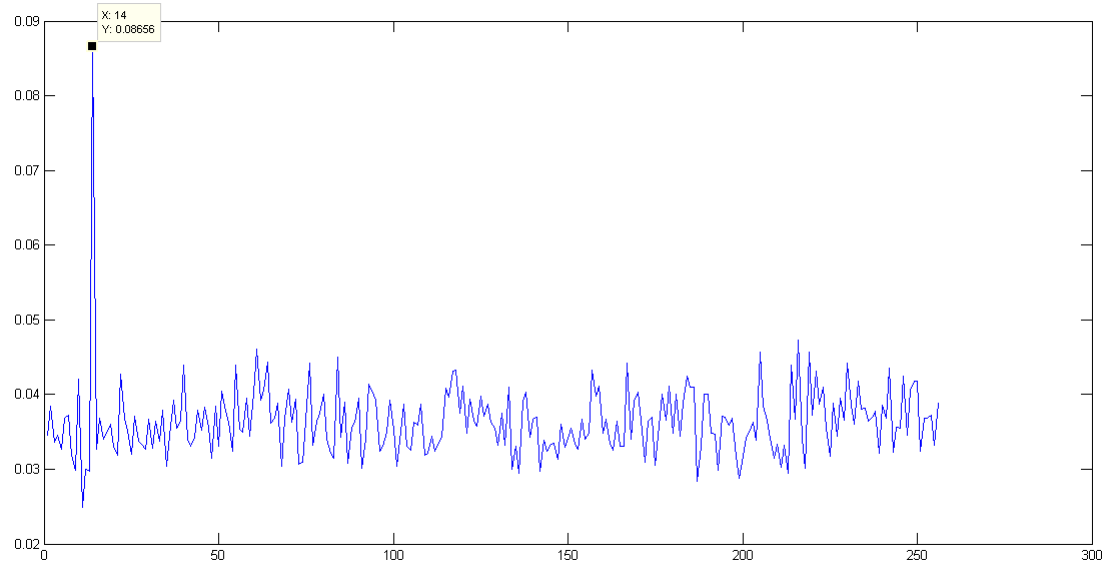

Figure A.5: DPA Attack into First Key with 10000 traces

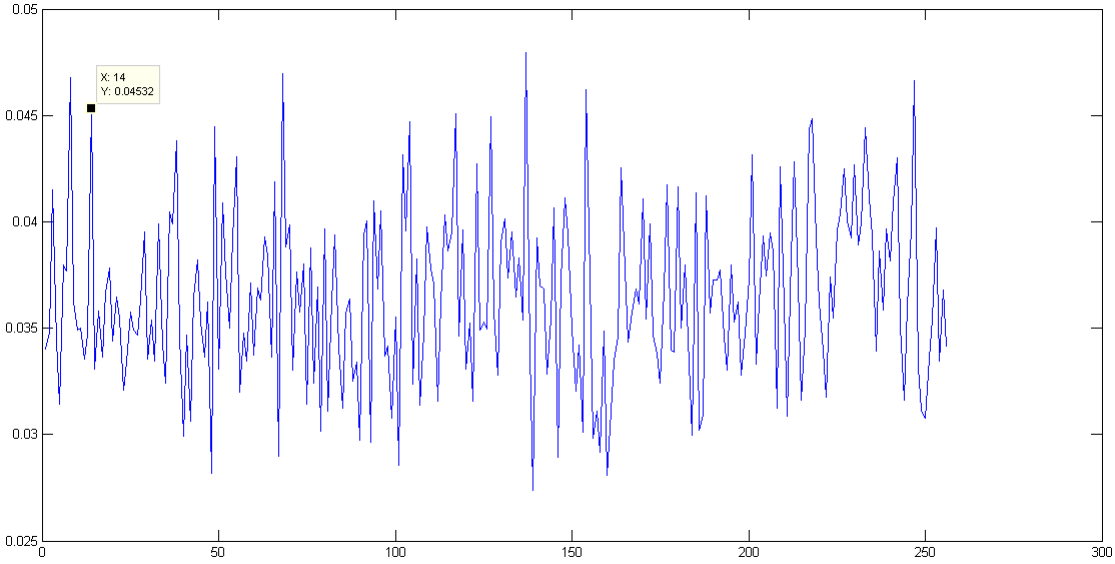

Figure A.6: CPA Attack into First Key with 10000 traces- Number of Bits used by Model : 1

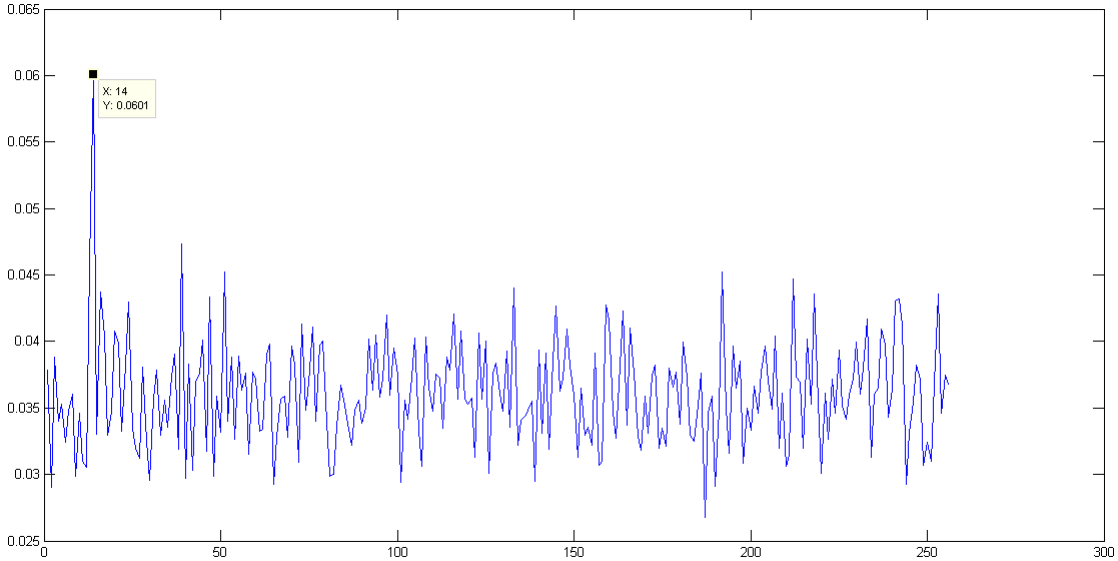

Figure A.7: CPA Attack into First Key with 10000 traces- Number of Bits used by Model : 2

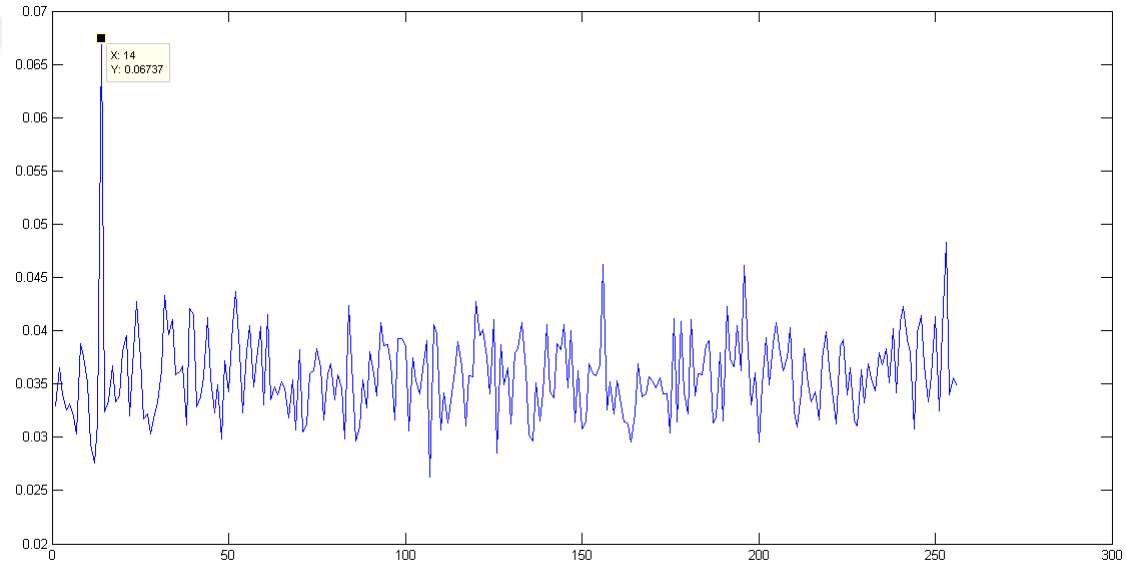

Figure A.8: CPA Attack into First Key with 10000 traces- Number of Bits used by Model : 4

APPENDIX B

Classification of Quadratic 5 x 5 S-Boxes

Classes	Class Representation																													
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	25	24	27	26	29	28	31	30
2	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	21	20	23	22	26	27	24	25	31	30	29	28
3	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	21	20	23	22	28	29	30	31	25	24	27	26
4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	24	25	26	27	28	29	30	31	20	21	22	23
5	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	19	18	22	23	21	20	28	29	31	30	26	27	25	24
6	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	19	18	24	25	27	26	28	29	31	30	20	21	23	22
7	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	20	21	24	25	28	29	22	23	18	19	30	31	26	27
8	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	19	17	24	26	27	25	28	30	31	29	20	22	23	21
9	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	19	17	24	26	27	25	29	31	30	28	21	23	22	20
10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	20	22	24	26	28	30	19	17	23	21	27	25	31	29
11	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	20	22	24	26	28	30	31	29	27	25	23	21	19	17
12	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	18	19	22	23	20	21	28	29	30	31	27	26	25	24
13	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	18	19	24	25	26	27	28	29	30	31	21	20	23	22
14	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	19	18	20	21	23	22	24	25	27	26	29	28	30	31
15	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	19	18	20	21	23	22	26	27	25	24	31	30	28	29
16	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	19	18	22	23	21	20	28	29	31	30	27	26	24	25
17	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	19	18	24	25	27	26	28	29	31	30	21	20	22	23
18	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	20	21	18	19	22	23	24	25	28	29	27	26	31	30
19	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	20	21	18	19	22	23	26	27	30	31	25	24	29	28
20	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	20	21	22	23	18	19	24	25	28	29	31	30	27	26
21	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	20	21	24	25	28	29	18	19	22	23	27	26	31	30
22	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	20	21	24	25	28	29	22	23	18	19	31	30	27	26
23	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	17	20	21	24	25	28	29	30	31	26	27	23	22	19	18
24	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	18	17	19	24	26	25	27	28	30	29	31	21	23	20	22
25	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	18	19	17	20	22	23	21	24	26	27	25	29	31	30	28
26	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	18	19	17	20	22	23	21	28	30	31	29	25	27	26	24
27	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	18	19	17	24	26	27	25	28	30	31	29	21	23	22	20
28	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	20	17	21	18	22	19	23	24	28	25	29	27	31	26	30
29	0	1	2	3	4	5	6	7	8	9	10	11	13	12	15	14	16	20	17	21	24	28	25	29	18	22	19	23	27	31	26	30
30	0	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	12	13	14	15	24	25	26	27	28	29	30	31	20	21	22	23
31	0	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	12	13	15	14	22	23	21	20	28	29	31	30	26	27	25	24
32	0	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	12	13	15	14	24	25	27	26	28	29	31	30	20	21	23	22
33	0	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	12	14	15	13	20	22	23	21	24	26	27	25	28	30	31	29
34	0	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	12	14	15	13	20	22	23	21	28	30	31	29	24	26	27	25
35	0	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	12	14	15	13	24	26	27	25	28	30	31	29	20	22	23	21
36	0	1	2	3	4	5	6	7	8	9	10	11	16	17	18	19	12	14	15	13	24	26	27	25	29	31	30	28	21	23	22	20
37	0	1	2	3	4	5	6	7	8	9	11	10	14	15	13	12	16	18	20	22	17	19	21	23	24	26	29	31	27	25	30	28
38	0	1	2	3	4	5	6	7	8	9	11	10	14	15	13	12	16	18	20	22	19	17	23	21	24	26	29	31	25	27	28	30
39	0	1	2	3	4	5	6	7	8	9	11	10	14	15	13	12	16	18	22	20	19	17	21	23	24	26	31	29	25	27	30	28
40	0	1	2	3	4	5	6	7	8	9	11	10	14	15	13	12	16	18	24	26	17	19	25	27	20	22	29	31	23	21	30	28
41	0	1	2	3	4	5	6	7	8	9	11	10	14	15	13	12	16	20	18	22	19	23	17	21	24	28	27	31	25	29	26	30
42	0	1	2	3	4	5	6	7	8	9	11	10	14	15	13	12	16	20	22	18	23	19	17	21	24	28	31	27	29	25	26	30
43	0	1	2	3	4	5	6	7	8	9	11	10	14	15	13	12	16	24	18	26	19	27	17	25	20	28	23	31	21	29	22	30
44	0	1	2	3	4	5	6	7	8	9	12	13	14	15	10	11	16	24	18	26	28	20	30	22	17	25	21	29	31	23	27	19
45	0	1	2	3	4	5	6	7	8	10	12	14	11	9	15	13	16	20	19	23	22	18	21	17	24	31	29	26	25	30	28	27
46	0	1	2	3	4	5	7	6	8	9	12	13	14	15	11	10	16	18	24	26	20	22	29	31	21	23	27	25	19	17	28	30
47	0	1	2	3	4	5	7	6	8	9	12	13	14	15	11	10	16	18	24	26	22	20	31	29	21	23	27	25	17	19	30	28
48	0	1	2	3	4	5	7	6	8	9	12	13	14	15	11	10	16	18	24	26	22	20	31	29	23	21	25	27	19	17	28	30
49	0	1	2	3	4	5	7	6	8	9	12	13	14	15	11	10	16	18	26	24	22	20	29	31	21	23	25	27	17	19	28	30
50	0	1	2	3	4	5	7	6	8	9	12	13	14	15	11	10	16	24	18	26	20	28	23	31	21	29	17	25	19	27	22	30

Table B.1: Quadratic Classes of quadratix 5×5 permutations

Classes	Class Representation																															
51	0	1	2	3	4	5	7	6	8	9	12	13	14	15	11	10	16	24	26	18	28	20	23	31	21	29	25	17	27	19	22	30
52	0	1	2	3	4	5	7	6	8	9	12	13	14	15	11	10	16	24	26	18	28	20	23	31	29	21	17	25	19	27	30	22
53	0	1	2	3	4	5	7	6	8	10	12	14	16	18	21	23	9	13	11	15	24	28	27	31	25	30	29	26	20	19	17	22
54	0	1	2	3	4	5	7	6	8	16	10	18	12	20	15	23	9	24	11	26	13	28	14	31	25	17	27	19	29	21	30	22
55	0	1	2	3	4	5	7	6	8	16	10	18	12	20	15	23	9	24	11	26	14	31	13	28	25	17	27	19	30	22	29	21
56	0	1	2	3	4	5	7	6	8	16	10	18	12	20	15	23	9	24	13	28	14	31	11	26	17	25	21	29	22	30	19	27
57	0	1	2	3	4	5	7	6	8	16	10	18	12	20	15	23	9	24	13	28	14	31	11	26	25	17	29	21	30	22	27	19
58	0	1	2	3	4	5	7	6	8	16	10	18	12	20	15	23	9	26	13	30	11	24	14	29	17	27	21	31	19	25	22	28
59	0	1	2	3	4	5	7	6	8	16	10	18	12	20	15	23	9	26	13	30	14	29	11	24	17	27	21	31	22	28	19	25
60	0	1	2	3	4	5	8	9	6	7	12	13	14	15	10	11	16	19	17	18	20	23	27	24	21	22	28	31	29	30	26	25
61	0	1	2	3	4	5	8	9	6	10	11	7	16	28	19	31	12	14	15	13	20	22	25	27	18	29	30	17	24	23	26	21
62	0	1	2	3	4	5	8	9	6	10	11	7	16	28	19	31	12	14	15	13	20	22	25	27	29	18	17	30	23	24	21	26
63	0	1	2	3	4	5	8	9	6	10	11	7	16	28	19	31	12	14	15	13	21	23	24	26	18	29	30	17	25	22	27	20
64	0	1	2	3	4	5	8	9	6	10	11	7	16	28	19	31	12	14	15	13	21	23	24	26	29	18	17	30	22	25	20	27
65	0	1	2	3	4	5	8	9	6	10	16	28	7	11	31	19	12	14	20	22	13	15	27	25	17	30	29	18	21	26	23	24
66	0	1	2	3	4	5	8	9	6	10	16	28	7	11	31	19	12	14	20	22	15	13	25	27	17	30	29	18	23	24	21	26
67	0	1	2	3	4	5	8	9	6	10	16	28	7	11	31	19	12	14	21	23	15	13	24	26	20	27	25	22	18	29	17	30
68	0	1	2	3	4	5	8	9	6	16	10	28	13	27	15	25	7	20	12	31	11	24	14	29	18	22	23	19	17	21	26	30
69	0	1	2	3	4	5	8	9	6	16	10	28	13	27	15	25	7	31	12	20	14	22	11	19	23	24	18	29	17	30	26	21
70	0	1	2	3	4	5	8	9	6	16	10	28	15	25	13	27	7	20	14	29	12	31	11	24	21	17	18	22	19	23	26	30
71	0	1	2	3	4	6	8	10	5	12	16	25	7	13	28	22	9	15	17	23	11	14	29	24	26	20	21	27	30	19	31	18
72	0	1	2	3	4	6	8	10	5	12	16	25	7	13	28	22	9	15	24	30	11	14	20	17	27	21	29	19	31	18	23	26
73	0	1	2	3	4	6	8	10	5	12	16	25	13	7	22	28	9	14	19	20	15	11	27	31	24	23	21	26	18	30	17	29
74	0	1	2	4	3	8	16	28	5	10	25	17	18	23	31	29	6	20	13	24	19	11	9	22	27	7	14	21	26	12	30	15
75	0	1	2	4	3	8	16	28	5	10	26	18	17	20	31	29	6	21	24	12	22	15	25	7	14	19	13	23	9	30	27	11

Table B.2: Quadratic Classes of quadratix 5×5 permutations