ANALYSIS OF BOOLEAN FUNCTIONS WITH RESPECT TO WALSH SPECTRUM

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

BY

ERDENER UYAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CRYPTOGRAPHY

SEPTEMBER 2013

Approval of the thesis:

ANALYSIS OF BOOLEAN FUNCTIONS WITH RESPECT TO WALSH SPECTRUM

submitted by **ERDENER UYAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy 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** _____

Assoc. Prof. Dr. Ali Doğanaksoy
Supervisor, **Department of Mathematics, METU** _____

Examining Committee Members:

Prof. Dr. Ersan Akyıldız
Department of Mathematics, METU _____

Assoc. Prof. Dr. Ali Doğanaksoy
Department of Mathematics, METU _____

Prof. Dr. Ferruh Özbudak
Department of Mathematics, METU _____

Assist. Prof. Dr. Zülfükar Saygı
Department of Mathematics, TOBB ETU _____

Dr. Fatih Sulak
Department of Mathematics, Atılım University _____

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: ERDENER UYAN

Signature :

ABSTRACT

ANALYSIS OF BOOLEAN FUNCTIONS WITH RESPECT TO WALSH SPECTRUM

Uyan, Erdener

Ph.D., Department of Cryptography

Supervisor : Assoc. Prof. Dr. Ali Doğanaksoy

September 2013, 60 pages

Boolean functions appear in various scientific disciplines including coding theory, combinatorics, complexity theory, cryptography, graph theory, etc. In cryptography, the design and analysis of Boolean functions possessing a range of cryptographic characteristics has often been the focus of attention. A productive ground of research for most of these cryptographic characteristics is Walsh spectrum, one of the most common representations of a Boolean function. This thesis presents an analysis of Boolean functions with respect to Walsh spectrum. The research is mainly devoted to the problem of determining the existence, construction and enumeration of n-variable Boolean functions having an arbitrary value, ω, appearing a certain number of times, s, in their Walsh spectrum. The thesis develops a new framework for the solution of this problem with parameters n, ω and s. Complete classification of Boolean functions of up to 6-variables is obtained within this framework. In higher dimensions, proof of existence by construction, several explicit formulas and bounds for various ω and s values are devised. On the other hand, the use of affine equivalence and the local connectivity is discussed. A new affine invariant property and an algorithm for computing the sizes of equivalence classes are introduced.

Keywords: Boolean functions, Walsh spectrum, spectral distribution, counting, affine equivalence

ÖZ

BOOLE FONKSİYONLARININ WALSH SPEKTRUMLARINA GÖRE ANALİZİ

Uyan, Erdener

Doktora, Kriptografi Bölümü

Tez Yöneticisi : Doç. Dr. Ali Doğanaksoy

Eylül 2013, 60 sayfa

Boole fonksiyonları kodlama teorisi, kombinatorik, karmaşıklık teorisi, kriptografi, çizge kuramı vs. gibi çeşitli bilimsel disiplinlerde ortaya çıkmaktadır. Kriptografide, kriptografik karakteristik çeşitliliği içeren Boole fonksiyonlarının tasarım ve analizi sık sık ilgi odağı olmuştur. Bu kriptografik karakteristiklerin çoğu için verimli bir araştırma alanı, Boole fonksiyonlarının en sık rastlanan gösterimlerinden biri olan Walsh spektrumdur. Bu tez Boole fonksiyonlarının Walsh spektruma göre bir analizini sunmaktadır. Araştırma temel olarak Walsh spektrumunda belirli bir s sayısı kadar gözüken rastgele bir ω değerine sahip n değişkenli Boole fonksiyonlarının varlığı, yapılandırılması ve sayılmasının belirlenmesi problemine adanmıştır. Tez bu problemin çözümü için n, w ve s parametreleriyle yeni bir çerçeve geliştirmektedir. Bu çerçeve dahilinde 6 değişkene kadar Boole fonksiyonlarının tam sınıflandırılması elde edilmiştir. Daha yüksek boyutlarda, yapılandırma yöntemiyle ispat, birkaç açık formül ve sınır bulunmuştur. Diğer taraftan, afin denkliğin kullanılması ve lokal bağlantısallık ele alınmıştır. Yeni bir afin değişmez ve denklik sınıflarının boyutlarını hesaplamak için bir algoritma sunulmuştur.

Anahtar Kelimeler: Boole fonksiyonlar, Walsh spektrum, spektral dağılım, sayma, afin denklik

To My Family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Assoc. Prof. Dr. Ali Doğanaksoy for supervising this thesis.

I also want to thank Dr. Çağdaş Çalık for his academic collaboration as well as his friendship during the whole process.

I wish to express my sincere appreciation to Assist. Prof. Dr. Zülfükar Saygı, Dr. Elif Yıldırım Saygı, Dr. A. Nurdan Saran, thesis committee members: Prof. Dr. Ferruh Özbudak and Prof. Dr. Ersan Akyıldız, and all friends at METU Institute of Applied Mathematics for their valuable contributions and feedback.

I also would like to thank administrative and academic staff of METU Department of Modern Languages for their support throughout my undergraduate and graduate studies.

I am deeply grateful to Aycan Yılmaz for her patience, help, encouragement and being with me all the way.

Last but not the least, I am happy to thank my dear family for their endless support, love and understanding.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 General Overview

Cryptography offers a variety of scientific research areas, each of which possess its unique set of problems. In most of these areas, especially those related to symmetric-key cryptographic systems, it is often a common practice to make use of Boolean functions to devise, express and solve these problems.

Boolean functions, also known as switching functions are named after the English mathematician George Boole (1815-1864). In 1854 with his publication of "An Investigation into the Laws of Thought"[3], Boole investigated the reasoning processes of the mind in mathematical language and he, in most simplistic terms, symbolized logic in terms of a new algebra. However, it was not before 1938, when Shannon employed this algebra in his paper "A symbolic analysis of relay and switching circuits"[35], that the theory of Boolean functions got its fame. Today, Boolean functions are studied in various scientific disciplines such as coding theory, combinatorics, computational complexity theory, cryptography, graph theory, information theory, logic synthesis and switching circuit theory.

In cryptography, Boolean functions are seen as the most fundamental and practiced components of cryptographic systems. Researchers have been studying the design and analysis of Boolean functions for a very long time now. The present state into which cryptography has evolved through that time is shaped considerably by these studies. Synthesizing and analyzing Boolean functions that possess a range of cryptographic properties has been the primary focus of research for cryptography. Day in, day out, several algorithms and constructions to generate desirable Boolean functions are proposed or improved. This is in fact accompanied by the continuous progress in the development of cryptanalysis techniques, which force the designers to use better choices of Boolean functions to be used in ciphers.

Boolean functions are simply mappings that output 0 or 1 for each n-bit input. They are often assessed with, but not limited to, the following main cryptographic features; balancedness, algebraic degree, nonlinearity, correlation immunity, algebraic immunity and propagation criteria. In order a function to be balanced,

it should produce each of its outputs in equal number of times. A balanced Boolean function can avoid being statistically distinguishable. Furthermore, attacks like the linear cryptanalysis [24] impose Boolean functions to have high nonlinearity,– the minimum distance to the set of affine functions. As for the correlation immunity, it was introduced by Siegenthaler in correlation attacks [36][37], as a measure of the correlation between a function's outputs and some subset of its inputs. Functions with high correlation immunity have many applications in key stream generation process of stream ciphers. Apart from that, algebraic attacks [11], introduced by Courtois and Meier, brought the concept of algebraic immunity into discussion. Functions with high algebraic immunity and high algebraic degree are expected to resist these type of attacks. Finally, the propagation criterion of degree p ($PC(p)$), introduced by Preneel [31], is satisfied when the output of a Boolean function changes with probability of one half whenever i ($1 \leq i \leq p$) input bits are complemented. $PC(p)$ is essential to analyze the behavior of a Boolean function when inputs are modified and $PC(p)$ of higher degrees is required to prevent lower approximation type of attacks.

On the other hand, these expected features usually conflict with each other. For instance, a function with even number of inputs can not be both balanced and of maximum nonlinearity, or a function of maximal algebraic immunity can not have algebraic degree greater than half of the number of inputs. Likewise, a function satisfying high correlation immunity may still possess a linear structure. Therefore, a 'balance' must be achieved between these desirable but incompatible features. In other words, it is a challenge to determine the existence or to construct Boolean functions involving a good combination of these properties that can be used safely in cryptographic systems. Some of the constructions for such functions can be seen in the publications of Carlet et al. [7][8], Limniotis et al. [19], Sarkar and Maitra [33].

Counting Boolean functions with predetermined parameters is also an interesting challenge. Not only it gives an idea about the success ratio of choosing these functions by random search, but it also provides the ability to see whether the functions used in a cryptographic system are chosen from a set of large cardinality. Besides, it enables to determine the effect of setting extra conditions on the chosen function type by observing the changes in the number of suitable functions.

There are several ways to represent a Boolean function such as truth table, algebraic normal form and Walsh spectrum being the most common ones. The most informative representation about cryptographic features is in fact its Walsh spectrum, a vector of coefficients obtained by the Walsh-Hadamard transform. Other than being able to compute balancedness, nonlinearity and correlation immunity directly, exploring new characteristics of Walsh spectrum can be employed to solve more problems of constructing, verifying and enumerating Boolean functions having properties such as resiliency, as studied in [10] and [23], or algebraic immunity as studied in [8]. It can also help to obtain new bounds on perfectly balanced Boolean functions as shown in [20]. All of those works emphasize the importance of finding relations among the Walsh coefficients. Thus, it is necessary to investigate this highly interesting area of study in cryptography further.

1.2 Motivation

There are various open problems regarding the existence, construction and enumeration of Boolean functions possessing certain cryptographic features (cf.[30]). Especially, the problems of counting Boolean functions that meet specified criteria has often taken a considerable amount of attention in cryptography, mainly because of its importance in defining and setting the boundaries for their search space. Finding the number of functions having maximum nonlinearity for even number of inputs, known as bent functions, or counting maximum nonlinear balanced Boolean functions are such two important open problems.

However, solving these open problems is quite hard. This is primarily due to their computational complexity cost. When the number of variables increases, the size of the whole Boolean function set expands double exponentially. Thus, their analysis require huge amount of computational resources, exceeding the capability of today's computers. The time required to solve these types of computational complexity problems can be reduced either using high computational resources or better algorithms. It would be much better to have both, however, the focus should be on the latter,– to be able to discover better techniques and algorithms that lead to faster solutions.

The main objective of this thesis is to develop algorithms and techniques to determine the existence, to construct and to enumerate Boolean functions having an arbitrary value appearing a certain number of times in their Walsh spectrum.

This research is developed based on these motivations and has the following contributions to the existing literature. First of all, the existence of Boolean functions with s many zeros in the Walsh spectrum for some s is shown by providing a construction by concatenation method. Exact distributions of Boolean functions of up to 6 variables with respect to Walsh coefficients are given. Several other results and bounds, obtained by exploiting the Parseval's Equation, affine classes, local connectivity and other combinatorial observations are also presented. Moreover, a new invariant property beneficial for testing affine equivalence is introduced. Using this invariant, a new enumeration algorithm that counts the number of functions in an affine equivalence class is presented.

1.3 Outline of the Thesis

The thesis is divided into five chapters including this introduction chapter.

In Chapter 2, notations and definitions about Boolean functions that will be frequently used throughout the thesis are provided. Representations of Boolean functions, namely truth table, algebraic normal form and Walsh spectrum are discussed in details.

The details of the problems of interest are defined and a mathematical framework

is constructed in Chapter 3. Also, previous works on Walsh coefficients and nonlinearity are discussed within this framework and our further findings are explained in this chapter.

Chapter 4 contains the improvement of the solution to the main problem in Chapter 3 by using the idea of affine equivalence. Moreover, a new invariant property, which can be used for testing equivalence of functions, is introduced and then used for a new algorithm to count the number of functions in an affine class.

Chapter 5 concludes the thesis by summarizing the work, identifying the contributions and providing suggestions for further research.

CHAPTER 2

PRELIMINARIES

2.1 Introduction

This chapter provides a review of the background material necessary to follow this thesis. Representations of Boolean functions, namely the truth table, algebraic normal form and the Walsh spectrum will be given. The Walsh-Hadamard transform and its famous speed-up algorithm, the Fast Walsh transform will be explained.

2.2 Boolean Functions

Finite Field $\mathbb{F}_2$ and Boolean function

The finite field consisting of only two elements 0 and 1 is denoted by $\mathbb{F}_2$. Two operations defined on this field are addition modulo 2, denoted by $\oplus$, and multiplication modulo 2, denoted by $\cdot$, which may be omitted for simplicity. Addition and multiplication tables are as follows.

Table 2.1: Addition and multiplication tables in $\mathbb{F}_2$

$\oplus$	**0**	**1**
0	0	1
1	1	0

$\cdot$	**0**	**1**
0	0	0
0	0	1

The n-dimensional vector space over $\mathbb{F}_2$ is $\mathbb{F}_2^n$. Let $\alpha, x \in \mathbb{F}_2^n$, then $\alpha \cdot x$ is the inner product of vectors in $\mathbb{F}_2^n$.

An element $\alpha = (\alpha_0, \alpha_1, \cdots, \alpha_{n-1})$ of $\mathbb{F}_2^n$ can be identified with an integer modulo 2^n by

$$\tilde{\alpha} = \sum_{i=0}^{n-1} \alpha_i 2^{n-i-1}.$$

This identification provides the ordering of the vectors in $\mathbb{F}_2^n$ with respect to their corresponding integer values, called the *lexicographic ordering*, i.e. $\tilde{0} \preceq \tilde{1} \preceq \tilde{2} \preceq \cdots \preceq 2^n \tilde{-} 1$. Note that the tilde symbol will be omitted where there is no ambiguity.

A Boolean function is a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. There are 2^n elements in $\mathbb{F}_2^n$; hence the set of n-variable Boolean functions, denoted by $\mathcal{B}_n$ has cardinality 2^{2^n}.

Truth Table and Polarity Truth Table

The vector $[f(\tilde{0}), f(\tilde{1}), \cdots, f(2^n \tilde{-} 1)]$ is called the *truth table* of a Boolean function f, that is basically listing all outputs of f with respect to the lexicographic ordering of its inputs.

The vector $[(-1)^{f(\tilde{0})}, (-1)^{f(\tilde{1})}, \cdots, (-1)^{f(2^n \tilde{-} 1)}]$ is called the *polarity truth table* of f. It is also shown by $(-1)^f$, or equivalently $1 - 2f$.

The *support* of f is the set $\Omega_f = \{\alpha \in \mathbb{F}_2^n \mid f(\alpha) = 1\}$ and the *weight* of f, $wt(f)$, is the cardinality of the support, i.e. $wt(f) = |\Omega_f|$.

The sum of two Boolean functions f and g is the function corresponding to the sum of truth table values of f and g.

Table 2.2: Sum and product of two 2-variable Boolean functions

f	g	$f \oplus g$	$f \cdot g$
0	0	0	0
1	1	0	1
1	0	1	0
1	1	0	1

The distance between two Boolean functions, denoted by $d(f, g)$ is the weight of their sum.

$$d(f, g) = wt(f \oplus g)$$

The distance between a Boolean function f and a set of functions S is

$$d(f, S) = min_{g \in S}\{wt(f \oplus g)\}$$

Algebraic Normal Form and Algebraic Degree

Another representation of a Boolean function is a polynomial in the quotient ring $\mathbb{F}_2[x_0, ..., x_{n-1}]/(x_0^2 - x_0, ..., x_{n-1}^2 - x_{n-1})$, called the *algebraic normal form* (ANF), shown as below.

$$f(x) = f(x_0, \cdots, x_{n-1}) = c \oplus \bigoplus_{0 \le i \le n-1} a_i x_i \oplus \bigoplus_{0 \le i < j \le n-1} a_{ij} x_i x_j \oplus \cdots \oplus a_{01\ldots n-1} x_0 x_1 \ldots x_{n-1},$$

where $c, a_i \in \mathbb{F}_2$.

The size of the largest product term in ANF of f is called the *algebraic degree*, or simply the *degree* of f, and is denoted by $deg(f)$.

ANF is crucial primarily for the computation of algebraic degree and truth table can be transformed to the algebraic normal form by means of the following transform easily.

Let f be the truth table of a Boolean function. Then,

$$\mathrm{ANF}_f = f \cdot A_n,$$

where $A_n = A_1 \otimes A_{n-1}$ for $n>1$ and $A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

Here, $\otimes$ stands for the *kronecker product* which is defined for $m \times n$ matrix $A = (a_{ij})$ and $p \times q$ matrix $B = (b_{ij})$ as the $mp \times nq$ matrix made up of $m \times n$ blocks where the (i, j) block is $a_{ij}B$, which is $p \times q$ submatrix obtained by multiplying each entry of B with a_{ij}.

The set of Boolean functions of degree less than or equal to r for $0 \leq r \leq n$ is associated with the r^{th}-order Reed-Muller code $RM(r, n)$ in coding theory, i.e. $RM(r, n) = \{f(x) \mid f(x) \in \mathcal{B}_n, deg(f) \leq r\}$. $RM(0, n)$ is the repetition code consisting of only all ones and all zeros vector, i.e. $RM(0, n) = \{0, 1\}$. On the other hand, $RM(-1, n)$ is accepted to be the zero codeword, i.e. $RM(-1, n) = \{0\}$. The quotient set, denoted by $RM(r, n)/RM(s, n)$, refers to the set $\{f(x) + RM(s, n) \mid s<deg(f) \leq r\}$, or equivalently all cosets of $RM(s, n)$ in $R(r, n)$.

Affine and Linear Boolean Functions

A Boolean function $f(x)$ of degree at most one is called an *affine function*. ANF of f is as follows

$$f(x) = l \cdot x \oplus c = l_0x_0 \oplus l_1x_1 \oplus \cdots \oplus l_{n-1}x_{n-1} \oplus c \ ,$$

where $c \in \mathbb{F}_2$ and $l \in \mathbb{F}_2^n$. An affine function with the constant term $c = 0$ is called a *linear function*. The functions $f(x) = 0$ or $f(x) = 1$, is called a *constant* function. The set of all n-variable affine (linear) functions is denoted by $\mathcal{A}_n$ ($\mathcal{L}_n$). It can be observed that $|\mathcal{A}_n|=2 \cdot |\mathcal{L}_n| = 2^{n+1}$.

Walsh-Hadamard Transform and Walsh Spectrum

One of the essential tools to study Boolean functions is the Walsh-Hadamard transform, which is defined for an n-variable Boolean function f as

$$W_f(\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \alpha \cdot x}, \alpha \in \mathbb{F}_2^n. \tag{2.1}$$

The vector $[W_f(\tilde{0}), W_f(\tilde{1}), ..., W_f(2^n \tilde{-} 1)]$ is called the *Walsh spectrum* (WS) of a Boolean function f, respectively. Each component $W_f(\alpha)$ of a Walsh spectrum

is called a *Walsh coefficient*, whose magnitude indicates the correlation between f and the corresponding linear function.

Fast Walsh Transform

Walsh-Hadamard transform is performed faster by a butterfly type algorithm called *Fast Walsh transform* (FWT). The transformation from truth table to the Walsh spectrum can be done by FWT with $\mathcal{O}(n2^n)$ complexity [34]. In order to do so, the truth table is first converted to the polarity truth table. At each step i, 2^{n-i} consecutive blocks of length 2^i are processed. First half of the block elements is added to the other half of the block and form the first half of the block in the next step. For the other half of the next step, the second half of the current block is subtracted from the first half of the block.

Figure 2.1 demonstrates an example of FWT on 3-variable Boolean functions, where $[x_0\ x_1\ \cdots\ x_7]$ is the polarity truth table of a Booelan function f and at the end of third step the Walsh spectrum in terms of x_i's is given.

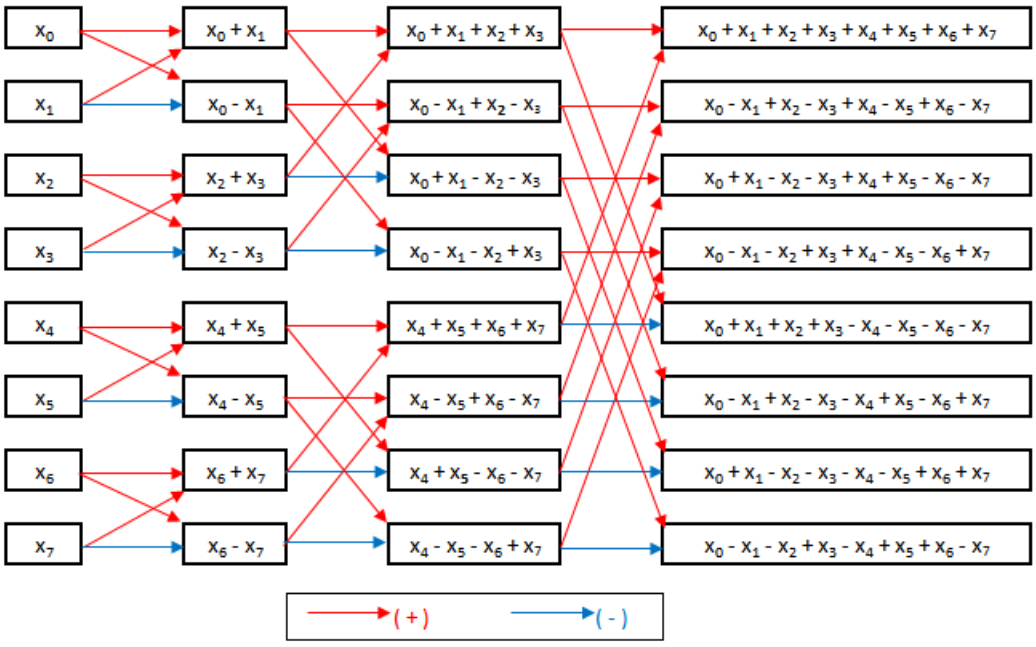

Figure 2.1: Fast Walsh transform for Boolean functions of 3 variables

Remark 2.1. Due to the recursive structure of FWT, Walsh spectra of 2^{n-i} Boolean functions of i-variable are formed at the end of i^{th} step of FWT ($i \leq n$). This makes constructions by concatenation possible.

Nonlinearity

Nonlinearity of a Boolean function is the minimum distance of a Boolean function f to the set of all affine functions.

$$nl(f) = d(f, \mathcal{A}_n) \tag{2.2}$$

The nonlinearity of f can be computed from the Walsh spectrum by

$$nl(f) = 2^{n-1} - \frac{1}{2} max_{\alpha \in \mathbb{F}_2^n} \{|W_f(\alpha)|\}. \tag{2.3}$$

Since nonlinearity is a measure of distance, its minimum value can be 0 and it is clear from 2.2 that $nl(f) = 0$ if and only if f is an affine function. In order to compute the maximum value of nonlinearity, the following fact, the Parseval identity, is used.

Fact 2.1. [21] Parseval Identity:

$$\sum_{\alpha \in \mathbb{F}_2^n} W_f(\alpha)^2 = 2^{2n} \tag{2.4}$$

To maximize $nl(f)$, $max_{\alpha \in \mathbb{F}_2^n} \{|W_f(\alpha)|\}$ value must be minimum possible. Since there are 2^n Walsh coefficients, each coefficient can be $\sqrt{\frac{2^{2n}}{2^n}} = \pm 2^{\frac{n}{2}}$ according to 2.4. Therefore, the maximum nonlinearity can be $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$, which is in fact possible if n is even.

Definition 2.1. [32][12] A Boolean function $f \in \mathcal{B}_n$ is called a *bent function*, if $W_f(\alpha) = \pm 2^{\frac{n}{2}}$ for all $\alpha \in \mathbb{F}_2^n$

Maximal nonlinearity is hence attained by bent functions. However, bent functions only exist when n is even.

CHAPTER 3

ANALYSIS OF BOOLEAN FUNCTIONS WITH SPECIFIED VALUES IN WALSH SPECTRUM

3.1 Introduction

This chapter presents our initial analysis on existence and enumeration of Boolean functions with specified values in their Walsh spectrum. The main problem of interest is discussed with a concise framework of parameters. Afterwards, the results obtained for several instances of these parameters are given. Finally, the chapter ends with a table of results. The work described here is based on the results of the publication "Counting Boolean functions with specified values in their Walsh spectrum" [40].

3.2 Problem

Our study on determining the existence and enumeration of the Boolean functions with specified number of Walsh values is initiated from the problem C3 in [30]. It asks whether there exists a Boolean function whose Walsh spectrum contains a specified number of zeros.

Problem 3.1. (**C3**[30]) Given an integer s, is there a function $f \in \mathcal{B}_n$ such that $\#\{\alpha \in \mathbb{F}_2^n \mid W_f(\alpha) = 0\} = s$?

This problem can be generalized by introducing a variable ω denoting a Walsh coefficient value and transforming the decision problem to a counting problem by asking the number of n-variable Boolean functions, whose Walsh spectrum contains a specified number s of a specified Walsh coefficient ω, exist in $\mathcal{B}_n$.

Problem 3.2. Given integers s and ω, how many functions f exist in $\mathcal{B}_n$ such that $\#\{\alpha \in \mathbb{F}_2^n| \ |W_f(\alpha)| = \omega\} = s$?

It should be noted that ω is the absolute value of a Walsh coefficient. This assumption has been made because the nonlinearity of a Boolean function is directly related to the magnitude of the coefficients in its Walsh spectrum. Thus, the sign of $W_f(\alpha)$ is omitted, and Walsh coefficients are considered with their

absolute values throughout the rest of the paper. Another advantage of omitting the signs is that the distribution of absolute values of Walsh spectrum remains invariant under affine transformations, which will be discussed in Chapter 4.

Problem 3.2 is important in the sense that it contains the distribution problem of the nonlinearities, or equivalently the weight distribution of first-order Reed-Muller codes. It is also related to the open problem of determining the number of bent functions in general and provides the classification and enumeration of Boolean functions in $\mathcal{B}_n$ with respect to their Walsh spectrum values. In order to follow a systematic approach, we first form a precise mathematical framework.

3.3 Framework

The variables involved in solution instances of Problem 3.2 are n,s and ω. Hence, the solutions are parametrized with respect to the 3-tuple (n,s,ω).

Definition 3.1. Let n,s and ω be nonnegative integers with $n \geq 1$ and ω, $s \leq 2^n$. We denote the set of Boolean functions such that $\#\{\alpha \in \mathbb{F}_2^n |\ |W_f(\alpha)| = \omega\} = s$ with $\mathcal{S}(n, s, \omega)$.

Definition 3.2. A **function distribution table** of $\mathcal{B}_n$ ($\boldsymbol{FDT}_n$) is a table whose entry at s^{th}row and ω^{th}column denotes the number of n-variable Boolean functions having Walsh coefficient ω appearing exactly s times in their Walsh spectrum.

The column headers of an FDT are the Walsh coefficients $|W_f(a)|$ considered as absolute values, whereas rows correspond to the number s of times a Walsh coefficient ω is observed in the Walsh spectrum of a function $f \in \mathcal{B}_n$. The template for FDT_n is given in Table 3.1.

Table 3.1: Function distribution table template

s \ ω	$0^\dagger$	2	4	...	$2^{n/2}$(n-even)	...	2^{n-1}	...	2^{n-k}	...	2^n-4	2^n-2	2^n
0				...		...		...		...			
1				...		...		...		...			Υ
2				...		...		...		...			
*				...		...		...		...			
*				...		...		...		...			
*				...		...		...		...			
2^n-2				...		...		...		...			
2^n-1	Υ			...		...		...		...			
2^n				...	ϑ	...		...		...			

(† Column pertaining to Problem 3.1)

Example 3.1. *$\mathcal{S}(n, 2^n-1, 0) = \mathcal{S}(n, 1, 2^n) = \mathcal{A}_n$ is the set of affine functions in $\mathcal{B}_n$. So $\Upsilon = |\mathcal{A}_n| = 2^{n+1}$ in Table 3.1.*

Example 3.2. *$\mathcal{S}(n, 2^n, 2^{n/2})$ is defined for even values of n and corresponds to the set of bent functions in $\mathcal{B}_n$. The cardinality ϑ of this set is known only up to $n = 8$ ([32], [18]).*

Remark 3.1. A function $f \in \mathcal{B}_n$ is counted in exactly one entry of each column of FDT_n. Thus, the sum of each column corresponds to $|\mathcal{B}_n| = 2^{2^n}$.

Remark 3.2. Let FDT_n^+ be FDT_n without the row $s = 0$. Then, a function $f \in \mathcal{B}_n$ is counted in r different entries of FDT_n^+, where r is the number of distinct Walsh coefficients (up to absolute values) in its spectrum.

3.4 Previous works in literature

The existence of Boolean functions with t nonzero Walsh coefficients, for particular values of t, was studied in [26] and [27]. These studies are related to $\mathcal{S}(n, s, 0)$ for $s = 2^n - t$ in the context of this paper

In [26], Pei and Qin showed that $|\mathcal{S}(n, s, 0)| = 0$ for $t = 2, 3, 5, 6$ and 7, where also the existence, i.e. $|\mathcal{S}(n, s, 0)|>0$, for $t = 4$ and $t = 8$ was shown by providing a construction for these parameters. However, there is no study concerning the number of those functions. On the other hand, in [27], Porwik gave the exact values of $|\mathcal{S}(n, 0, 0)|$ and $\sum_{s=1}^{2^n} |\mathcal{S}(n, s, 0)|$ up to $n = 5$, by exhaustive search.

In [9], Carlet and Mesnager gave a construction for a class of Boolean functions having a single zero in their Walsh spectrum for $(n \geq 10)$. This class belongs to the set $\mathcal{S}(n, 1, 0)$ and the work proves that $|\mathcal{S}(n, 1, 0)| \neq 0$ for $(n \geq 10)$.

Finally, Wu in [42] gave the distribution of Boolean functions with nonlinearity $nl(f) \leq 2^{n-2}$. This is similar to finding $|\mathcal{S}(n, s, \omega)|$ for all s and n such that $\omega \geq 2^{n-1}$ in our context.

3.5 Results

This section covers the results that have been obtained as our partial solutions to Problem 3.2. The results are partitioned into distinct cases of solution sets $\mathcal{S}(n, s, \omega)$.

3.5.1 $\mathcal{S}(n, s, \omega)$ for all s and ω such that $n \leq 5$

An exhaustive search on n-variable Boolean functions for $n \leq 5$ is performed in order to find $\mathcal{S}(n, s, \omega)$ for all s and ω. The results of these computations are compiled to form the corresponding FDT, all of which can be found in Appendix A.

Exhaustive search is infeasible for $n \geq 6$ as $\mathcal{B}_n$ grows exponentially in n. In order to find $\mathcal{S}(n, s, \omega)$ for $n = 6$, we will use the idea of affine equivalence in Chapter 4. However, our aim at this point is to understand the general case by spotting particular solutions for larger values of n by utilizing the following facts.

Fact 3.3. One bit change in truth table of a Boolean function f adds ± 2 to each component of $WS(f)$.

Fact 3.4. For any $\alpha \in \mathbb{F}_2^n$, $W_f(\alpha) \equiv 0$ *(mod 4)*, if $wt(f)$ is even and $W_f(\alpha) \equiv 2$ *(mod 4)* otherwise.

Fact 3.5. The distance between two affine functions, whose sum is non-constant, is 2^{n-1}.

Fact 3.6. The weight of the product of two affine functions, whose sum is non-constant, is 2^{n-2}.

3.5.2 $\mathcal{S}(n, s, \omega)$ for all s and n such that $\omega > 2^{n-1}$

In [42], it is shown that the number of Boolean functions with nonlinearity $nl(f)$ is $2^{n+1}\binom{2^n}{nl(f)}$ for $nl(f) < 2^{n-2}$. Here, we obtain this result with an alternative self-contained approach that we also benefit for proving other results such as the propositions 3.10 and 3.12.

Proposition 3.7. *Let* $\omega = 2^n - 2k$ *such that* $k < 2^{n-2}$. *Then,* $|\mathcal{S}(n, 1, \omega)| = 2^{n+1}\binom{2^n}{k}$ *and* $|\mathcal{S}(n, s, \omega)| = 0$ *for all* $s > 1$.

Proof. $k < 2^{n-2} \Rightarrow \omega > 2^{n-1}$. In order to count the functions in corresponding $\mathcal{S}(n, s, \omega)$, we start from affine Boolean functions and consider consecutive bit-flips in the truth tables of these functions. It is well known that affine functions contain $(2^n - 1)$ 0's and a single $\pm 2^n$ in their Walsh spectrum (see Example 3.1). These functions are located in the rightmost column of FDT_n. Using Fact 2.1, it is clear that $\mathcal{S}(n, s, \omega) = 0$ for $s > 1$ in that column. So the proposition holds for $\mathcal{S}(n, s, 2^n)$.

On the other hand, Fact 3.3 implies that a single bit change in the truth table of an affine function produces 2^n new functions with their largest Walsh coefficient values changing to $\pm(2^n - 2)$ and 0's being changed to ± 2. If we continue to flip truth table bits, we can go up to the point where magnitudes of changed 0's and 2^n collide. Fact 3.5 implies that this collision occurs at $(\frac{2^{n-1}}{2} = 2^{n-2})$. Until that point, k bit changes in the truth table of an affine function results in $\binom{2^n}{k}$ different functions at k distance to the starting affine function. This produces $2^{n+1}\binom{2^n}{k}$ functions that have the Walsh coefficient $\omega = 2^n - 2k$ in their Walsh spectrum.

Moreover, up to the point where changed 0's and 2^n collide, i.e. $k = 2^{n-2}$, the ones produced from 2^n stay single as none of the 0's reach their magnitude yet. In other words, for $\omega > 2^{n-1}$ and $s > 1$, $|\mathcal{S}(n, s, \omega)| = 0$, since the number of changed bits is less than 2^{n-2}. □

This result gives the exact number of n-variable Boolean functions for the right half of FDT_n, i.e. for the values of $|\mathcal{S}(n,s,\omega)|$ for all s and n such that $\omega{>}2^{n-1}$. The case where $\omega = 2^{n-1}$ will be discussed in Proposition 3.12.

3.5.3 $\mathcal{S}(n,s,\omega)$ for all n and for some s such that $\omega = 0$

In this section, the existence of elements in $\mathcal{S}(n,s,0)$, applicable to all n but limited to some s values, is proved by the following proposition.

Proposition 3.8. *Let $\mathcal{S}_m$ be defined as the set consisting of the values s such that $|\mathcal{S}(m,s,0)|{>}0$. Let $\zeta \in \mathcal{S}_m$ and $\eta \in \mathcal{S}_m \backslash \{2^m-1\}$. Then, for all $n \geq m+1$*

i. $\tilde{\zeta} = \zeta + \sum_{i=0}^{n-m-1} 2^{i+m} \in \mathcal{S}_n$

ii. $\tilde{\eta} = \eta \cdot 2^{n-m} \in \mathcal{S}_n$

Proof. The following constructions prove the existence of such $\tilde{\zeta}$ and $\tilde{\eta}$ in $\mathcal{S}_n$:

i. *Step 1.* Concatenate the Walsh Spectrum of a Boolean function $WS(f)$ to itself for an arbitrary $f \in \mathcal{B}_{n-1}$, and obtain the vector $[WS(f)\|WS(f)]$.
Step 2. Apply the last step of the Fast Walsh transform [34], i.e. compute $FWT_n([WS(f)\|WS(f)])$.
The output is a new Walsh spectrum in n^{th} dimension having 2^{n-1} more zeroes than $WS(f)$.

ii. If the procedure in (i) is applied to any $WS(f)$ such that $f \notin \mathcal{A}_{n-1}$ by concatenating it to the Walsh spectrum of an affine Boolean function $l \in \mathcal{A}_{n-1}$, the result is a Walsh spectrum with the number of zeros doubled.

Note that while choosing η, the value 2^m-1 is excluded from $\mathcal{S}_m$ since it corresponds to affine functions. Concatenation of two affine functions clearly cannot result in doubling of the s value, but a new spectrum with either 2^n-1 or 2^n-4 zero Walsh values.

□

Using $s=0$ column of FDT_5, it is known that

$$\mathcal{S}_5 = \{4,6,7,8,9,10,11,12,14,16,19,22,24,28,31\}\ .$$

Once, $\tilde{\zeta}$ and $\tilde{\eta}$ values are obtained for all ζ and η in $\mathcal{S}_5$ by the above constructions, a subset

$$\tilde{\mathcal{S}}_6 = \{8,12,14,16,18,20,22,24,28,32,36,38,$$
$$39,40,41,42,43,44,46,48,51,54,56,60,63\}$$

of $\mathcal{S}_6$ is formed. Similarly, $\tilde{\mathcal{S}}_6$ can now be employed to produce a subset for $\mathcal{S}_7$. Therefore, both procedures should be used recursively and in combination to obtain new functions having $\tilde{\zeta}$ and $\tilde{\eta}$ zeroes in their Walsh spectrum for higher values of n. Table 3.2 shows the number of s values generated this way and the percentages taken with respect to 2^n, number of all possible s values. Later, this result will be improved according to FDT_6 values in Chapter 4.

Table 3.2: Number of s values generated by Proposition 3.8 using $\mathcal{S}_5$

n	Number of all s (2^n)	$\#s$ generated	Percentage	$\lvert\mathcal{S}_n\rvert$
5	32	15	46.88% (15/32)	15
6	64	25	39.06% (25/64)	unknown[†]
7	128	51	39.84% (51/128)	unknown
8	256	69	26.95%(69/256)	unknown
9	512	87	16.99% (87/512)	unknown

[†] This number is going to be known in Table 4.2

3.5.4 Other Results

Proposition 3.9. *Let* $\sigma = \frac{2^{2n}}{\omega^2}$. *Then* $|\mathcal{S}(n,s,w)| = 0$ *for* $\omega > 2^{n/2}$ *and* $s > \sigma$.

Proof. Let $WS(f) = [W_f(\alpha_0), W_f(\alpha_1), ..., W_f(\alpha_\sigma), W_f(\alpha_{\sigma+1}), ..., W_f(\alpha_{2^n-1})]$ be a Walsh spectrum containing the value ω maximum number of times, say σ. Without loss of generality assume that the first σ values are ω. Then using 2.1,

$$\sigma\omega^2 + (W_f(\alpha_{\sigma+1})^2 + ... + W_f(\alpha_{2^n-1})^2) = 2^{2n}$$

$$\Leftrightarrow \sigma\omega^2 \leq 2^{2n} \Leftrightarrow \sigma \leq \frac{2^{2n}}{\omega^2}$$

Thus, σ is an upper bound for the maximum possible s value of a Walsh coefficient ω in a Walsh spectrum. Since $\sigma = 2^n$ when $\omega = 2^{n/2}$ (for even n), the proposition provides a bound for all $\omega > 2^{n/2}$, which covers a significant number of columns of FDT_n. □

Definition 3.3. A **multiset** $\mathcal{M}$ is a generalization of a set, in which repetition of the same element is allowed. Number of occurrences of an element m in $\mathcal{M}$ is called its **multiplicity**, denoted by $\mu(m)$.

Proposition 3.10. *Let the multiset of all Walsh coefficients of all functions in* $\mathcal{B}_n$ *be denoted by* $\mathcal{W}_n$, *which has cardinality* 2^{2^n+n}. *Let* $0 \neq |W_f(\alpha)| = \omega \in \mathcal{W}_n$, *then the multiplicity of* ω *is*

$$\mu(\omega) = 2^{n+1}\binom{2^n}{2^{n-1} + \frac{\omega}{2}}, \tag{3.1}$$

and $\mu(0) = 2^n\binom{2^n}{2^{n-1}}$.

Proof. Fix an arbitrary $a \in \mathbb{F}_2^n$. Then, $W_f(\alpha) = \omega$ if and only if f is at distance $2^{n-1} + \frac{w}{2}$ to the linear function $\alpha \cdot x$. The number of such functions is $\binom{2^n}{2^{n-1}+\frac{\omega}{2}}$, which can be obtained by making $2^{n-1} + \frac{\omega}{2}$ changes to TT of the linear function $a \cdot x$. Also, $W_f(\alpha) = -w$ occurs when the number of changes is $2^{n-1} - \frac{\omega}{2}$. Since $\binom{2^n}{2^{n-1}+\frac{\omega}{2}} = \binom{2^n}{2^{n-1}-\frac{\omega}{2}}$, the number of Boolean functions with $|W_f(\alpha)| = \omega$ is $2\binom{2^n}{2^{n-1}+\frac{\omega}{2}}$. This fact holds for any $\alpha \in \mathbb{F}_2^n$, so the total number of Walsh coefficient ω in $\mathcal{W}_n$ is $2^{n+1}.\binom{2^n}{2^{n-1}+\frac{\omega}{2}}$.

For $\omega = 0$, similar reasoning holds. However, multiplicity is counted twice since $\omega = -\omega$ in this case. So the total number should be halved for $\mu(0)$. □

Corollary 3.11. *Let the header column of* FDT_n *be the vector* S*, i.e.* $S = [1, 2, \cdots, 2^n]$*, and the column corresponding to an arbitrary* ω *be denoted by* Ω*. Then,*

$$\mu(\omega) = S \cdot \Omega$$

Proof. This is clear from the definition of $FDT_n(\omega)$ and $\mu(\omega)$. □

Definition 3.4. Let $f, g \in \mathcal{B}_n$. f is called the **complement** of g or vice versa if $f \oplus g = 1$. A set of Boolean functions is called **complement-free** if no two of its elements are complements of each other .

Proposition 3.12. *Let* $\omega = 2^{n-1}$*. Then,*

i. $|\mathcal{S}(n, s, \omega)| = 0$ *for* $s \geq 5$

ii. $|\mathcal{S}(n, 4, \omega)| = 2\binom{2^n}{3}$

iii. $|\mathcal{S}(n, 3, \omega)| = 0$

iv. $|\mathcal{S}(n, 2, \omega)| = \Psi$

v. $|\mathcal{S}(n, 1, \omega)| = \Phi$

vi. $|\mathcal{S}(n, 0, \omega)| = 2^{2^n} - [\Phi + \Psi + 2\binom{2^n}{3}]$,

where $\Phi = \mu(\omega) - \left[2\beta + 4 \cdot 2\binom{2^n}{3}\right]$ *and* $\Psi = \left[\binom{2^{n+1}}{2} - 2^n\right]\binom{2^{n-1}}{2^{n-2}} - \binom{4}{3}2\binom{2^n}{3}$.

Proof. A Walsh spectrum containing $\omega = 2^{n-1}$ can only be obtained by making 2^{n-2} changes to the truth table of an affine function. Therefore, we are interested in the number of functions produced in this way. Since the distance between two complement-free affine functions is 2^{n-1}, a function can be obtained from more than one affine function. In order for two affine functions to meet at the same function, the changes must be done to the bits that differ in their truth tables. So, $\binom{2^{n-1}}{2^{n-2}}$ functions, which are at distance 2^{n-2} to both of these affine functions can be produced.

Given two complement-free affine functions f_1 and f_2, a third one f_3 can be chosen so that all three can meet at a Boolean function g that is at 2^{n-2} distance to all. For this to happen, there should be 2^{n-2} truth table positions in which f_2 and f_3 agree and f_1 does not. These positions correspond to the support of the function $(f_1 + f_2)(f_1 + f_3)$, whose weight is 2^{n-2} from Fact 3.6. Hence, when these 2^{n-2} positions are changed in f_1, the resulting function gets 2^{n-2} many bits closer to both f_2 and f_3. After this point, g can be transformed to f_2 and f_3 by making 2^{n-2} changes.

Now, we will show that when the affine functions f_1, f_2, f_3 meet at a function g, a fourth one f_4 always exists and is of the form $f_4 = f_1 + f_2 + f_3 + 1$, which is also at distance 2^{n-2} to g. The existence of the fourth function can be justified by showing that there are 2^{n-2} truth table positions, in which f_1, f_2 and f_3 agree and f_4 does not. These positions correspond to the support of the function $(f_1 + f_2 + 1)(f_1 + f_3 + 1)$, whose weight is also 2^{n-2}. When the truth tables of f_1, f_2, f_3 agree at a point, f_4 has the complement of their values at that point. Because, when $f_1 = f_2 = f_3 = 0$, f_4 becomes 1 and when $f_1 = f_2 = f_3 = 1$, f_4 becomes 0. Note that g is in fact $f_1f_2+f_1f_3+f_2f_3$, since $g+f_1 = (f_1+f_2)(f_1+f_3)$, $g + f_2 = (f_1 + f_2)(f_2 + f_3)$, $g + f_3 = (f_1 + f_3)(f_2 + f_3)$ and $g + f_4 = (f_1 + f_2 + 1)(f_1+f_3+1)$ are all products of two affine functions and have weights 2^{n-2} (Fact 3.6).

Thus, the inevitable existence of the fourth function proves that there exist no functions in $\mathcal{S}(n, 3, \omega)$; hence (iii) is proven.

The number of cases when four affine functions produce the same function is the number of complement-free four-tuples $\{f_1, f_2, f_3, f_4\}$ such that $f_4 = f_1 + f_2 + f_3 + 1$. Since f_4 is determined by f_1, f_2, f_3, it is enough to choose first three functions. So, f_1 is chosen from 2^{n+1} affine functions in 2^{n+1} ways. f_2 is chosen in $(2^{n+1} - 2)$ ways from all affine functions excluding f_1 and $f_1 \oplus 1$. f_3 is chosen in $(2^{n+1} - 4)$ ways from all affine functions excluding f_1, f_2, $f_1 \oplus 1$ and $f_2 \oplus 1$. This makes $2^{n+1}(2^{n+1} - 2)(2^{n+1} - 4)$ ways to choose f_1, f_2 and f_3. However, we should divide this number by 4, since each quadruple of functions $\{f_1, f_2, f_3, f_4\}$ is counted 4 times with this method. Moreover, depending on the choice of order of the first three functions, it should also be divided by 3!, which makes

$$\frac{2^{n+1}(2^{n+1} - 2)(2^{n+1} - 4)}{4 \cdot 3!} = \frac{2^{n+1}(2^n - 1)(2^n - 2)}{3!} = 2\binom{2^n}{3}$$

Thus, (ii) is proven.

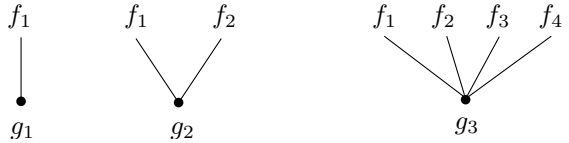

Figure 3.1: Functions at 2^{n-2} distance to affine functions.

Recall that $\binom{2^{n-1}}{2^{n-2}}$ functions can be produced that are at distance 2^{n-2} to two affine functions. Two complement-free affine functions can be chosen in $\binom{2^{n+1}}{2}-2^n$ ways. This makes $\left[\binom{2^{n+1}}{2}-2^n\right]\binom{2^{n-1}}{2^{n-2}}$ new functions in total. In order to count the functions produced by exactly two affine functions, those generated by four affine functions must be subtracted. Such functions are counted $\binom{4}{2}=6$ times. Therefore, excluding them we have $\left[\binom{2^{n+1}}{2}-2^n\right]\binom{2^{n-1}}{2^{n-2}}-\binom{4}{2}2\binom{2^n}{3}$ functions in $\mathcal{S}(n,2,\omega)$. Hence, (iv) is proven.

The number of functions that are at 2^{n-2} distance to exactly one affine function can be found using Corollary 3.11.

$$\mu(\omega)=|\mathcal{S}(n,1,\omega)|+2|\mathcal{S}(n,2,\omega)|+4|\mathcal{S}(n,4,\omega)|$$

$$\Rightarrow |\mathcal{S}(n,1,\omega)|=\mu(\omega)-2|\mathcal{S}(n,2,\omega)|-4|\mathcal{S}(n,4,\omega)|$$

Thus, case (v) is computed. The last case (vi) is the number of remaining functions in $\mathcal{B}_n$. □

Proposition 3.13. *Let $n \geq 2$.*

$$|\mathcal{S}(n,0,2)| \geq 2^{2^n-1},$$

where the equality holds only for $2 \leq n \leq 5$.

Proof. This is equivalent to proving that $\sum_{s=1}^{2^n}|\mathcal{S}(n,s,2)| \leq 2^{2^n-1}$, namely the number of functions having 2 in WS. By Fact 2.1, if there exists a Walsh coefficient $\omega_i > \sqrt{\frac{2^{2n}}{2^n}} = 2^{n/2}$, then there must be at least one $\omega_j < 2^{n/2}$. Moreover, using Fact 3.4, we know that for n-variable functions having odd weight with $2 \leq n \leq 5$, the only Walsh coefficient value ω satisfying $\omega < 2^{n/2}$ and $\omega \equiv 2 \pmod 4$ is 2, since $2^{n/2} < 6$ for $2 \leq n \leq 5$. In other words, functions having 2 in WS are exactly the odd weight Boolean functions, which in total makes 2^{2^n-1} functions.

For $n \geq 6$, there exist functions with $2 < \omega < 2^{n/2}$ and $\omega \equiv 2 \pmod 4$, since $2^{n/2} > 6$, thus keeping the number of $|\mathcal{S}(n,0,2)|$ below the number of all odd weight functions. □

Table 3.3: FDT_n

s \ $\lvert\omega\rvert$	0	2	...	$2^{n/2}$ (even-n)	...	2^{n-1} (3.12)[42]	$2^n - 2k$ $(0 \leq k < 2^{n-2})$ (3.7) [42]
0	?	$\geq 2^{2^n-1}$ (3.13)	....	?	....	$2^{2^n} - \left[\Phi + \Psi + 2\binom{2^n}{3}\right]$	$2^{2^n} - 2^{n+1}\binom{2^n}{k}$
1	# [9]	?	...	?	...	Φ	$2^{n+1}\binom{2^n}{k}$
2	?	?	...	?	...	Ψ	0
3	?	?	...	?	...	0	0
4	?	?	...	?	...	$2\binom{2^n}{3}$	0
.	?	?	...	?	...	0	0
$\hat{s} \in \mathcal{S}_n$ (3.8)	# (3.8)	?	...	?	...	0	0
.	?	?	...	?	...	0	0
σ (3.9)	?	?	....	?	# (3.9)	0	0
.	?	?	....	?	0	0	0
$2^n - 7$	0 [26]	?	...	?	0	0	0
$2^n - 6$	0 [26]	?	...	?	0	0	0
$2^n - 5$	0 [26]	?	...	?	0	0	0
$2^n - 4$	$2\binom{2^n}{3}$ (3.12)	?	...	?	0	0	0
$2^n - 3$	0 [26]	0	...	?	0	0	0
$2^n - 2$	0 [26]	0	...	?	0	0	0
$2^n - 1$	2^{n+1}	$2^{n+1}2^n$	...	0	0	0	0
2^n	0	0	0	ϑ ($n = 8$ [18])	0	0	0
$\mu(\omega)$(3.10)	$2^n\binom{2^n}{2^{n-1}}$					$2^{n+1}\binom{2^n}{2^{n-1}+\frac{\omega}{2}}$	

$$\Phi = \mu(\omega) - \left[2\beta + 4\binom{2^n}{3}\right], \qquad \Psi = \left[\binom{2^{n+1}}{2} - 2^n\right]\binom{2^{n-1}}{2^{n-2}} - \binom{4}{3}2\binom{2^n}{3}$$

CHAPTER 4

ANALYSIS OF BOOLEAN FUNCTIONS WITH RESPECT TO WALSH SPECTRUM USING EQUIVALENCE CLASSES

4.1 Introduction

This chapter contains the utilization of equivalence classes to improve results in Chapter 3. Firstly, the importance of affine transformations, various facts and known affine invariant properties are going to be explained. After this preliminary information, construction of FDT_6, included in [41], will be explained. A new invariant property that can be used for testing equivalence of functions will be introduced. This property will then be used as the core of a counting algorithm that gives the number of functions in an equivalence class. The algorithm developed in this chapter is the first known systematic algorithm to quantify this number.

4.2 Affine Transformations and Equivalence Classes

Given the fact that $\mathcal{B}_n$ contains 2^{2^n} functions, it isn't manageable to analyze each function even for relatively small values of n. This bound is as small as 6 due to today's computation limits. In other words, $\mathcal{B}_6$ is not small enough to be able to list all of its elements. Therefore, a convenient method would be to partition this large set into equivalence classes by defining a suitable equivalence relation. Choosing representatives for each class, this large set can be reduced to the number of classes which makes the analysis easier. The most commonly used equivalence relation is the affine transformations defined as follows.

Definition 4.1. A mapping $\mathcal{B}_n \to \mathcal{B}_n$, $g \mapsto f$ such that

$$f(x) = g(Ax \oplus a) \oplus bx \oplus c \ ,$$

where A is a nonsingular binary $n \times n$ matrix, $a, b \in \mathbb{F}_2^n$ and $c \in \mathbb{F}_2$, is called an *affine transformation*.

This definition states that an affine transformation is a composition of a linear mapping, $\mathbb{F}_2^n \to \mathbb{F}_2^n$, *i.e.*$Ax$, a variable complementation, i.e. $x \oplus a$, and an affine shift, i.e. $g(x) \oplus bx \oplus c$. Applying all possible affine transformations to a Boolean function produces a class of Boolean functions that possess similar characteristics, which will be discussed in Section 4.3.

Definition 4.2. Two Boolean functions f and g are called *affine equivalent* if there exists an affine transformation from f to g. Moreover, such f and g are said to be in the same *equivalence class*.

Equivalence is also generalized in terms of Reed-Muller codes as follows.

Definition 4.3. Two Boolean functions $f, g \in RM(r,n)/RM(s,n)$ are called *equivalent* over $RM(s,n)$ if there exists a nonsingular binary $n \times n$ matrix A and $a \in \mathbb{F}_2^n$ such that $f(x) = g(Ax \oplus a)\ mod\ RM(s,n)$.

The case $s = 1$ corresponds to the definition of affine equivalence (Def. 4.2), since $RM(1,n)$ contains only affine functions. If $s = 0$, then $b = 0$, i.e. $f(x) = g(Ax \oplus a) \oplus c$, and the functions f and g are said to be equivalent over $RM(0,n)$. If the functions are equivalent over $RM(-1,n)$, i.e. $s = -1$, then b and c are both 0.

Complete classification for $\mathcal{B}_n$ with respect to equivalence classes is known only up to $n = 6$, whereas only the total number[16] and classes in $RM(3,7)/RM(1,7)$ [5] is known for $n = 7$. Table 4.1 shows the number of equivalence classes of $\mathcal{B}_n$.

Table 4.1: Number of equivalence classes for small values of n. [38]

n	**Number of classes**
1	1
2	2
3	3
4	8
5	48 [2]
6	150357 [22]
7	33791473207774085 48 [16]

4.3 Invariant Properties

Many cryptographic properties are invariant under affine transformations such as nonlinearity, algebraic degree, algebraic immunity, or more importantly for this study, the frequency distribution of absolute values of Walsh coefficients. In [6], Braeken et al. surveys the invariant properties in details. Below, we give the definition of "invariant" and state the invariant properties relevant to our study.

Definition 4.4. A mapping M from $RM(r,n)/R(s,n)$ to a set is called an *invariant* of $R(r,n)/R(s,n)$, if for any two equivalent functions $f(x), g(x) \in R(r,n)/R(s,n)$, $M(f) = M(g)$ holds.

Proposition 4.1. *([28] 8.3) For two equivalent functions f and g such that $f(x) = g(Ax \oplus a) \oplus bx \oplus c$, i.e. equivalent over $RM(1,n)$,*

$$W_f(w) = (-1)^{(w\oplus b)A^{-1}a\oplus c} W_g((w \oplus b)A^{-1}). \tag{4.1}$$

Proof.

$$W_f(w) = \sum_{w\in\mathbb{F}_2^n} (-1)^{f(x)\oplus wx} = \sum (-1)^{g(Ax\oplus a)\oplus bx\oplus c\oplus wx}$$

Substituting $(w \oplus b)$ with w' and $(Ax \oplus a)$ with x',

$$W_f(w' \oplus b) = \sum (-1)^{g(x')\oplus w'A^{-1}(x'\oplus a)\oplus c}$$

Now, substituting $(w'A^{-1})$ with w'',

$$W_f(w''A \oplus b) = \sum (-1)^{g(x')\oplus w''x'\oplus w''a\oplus c} = (-1)^{w''a\oplus c} W_g(w'')$$

Finally, changing $(w''A + b)$ back to w and replacing w'' with $(w \oplus b)A^{-1}$, the result is obtained. □

Remark 4.1. Proposition 4.1 implies that the absolute value of Walsh coefficient of $f(x)$ at w is equal to the absolute value of Walsh coefficient of $g(x)$ at v such that $w = Av \oplus b$. Since A is a nonsingular binary matrix, the distribution of the absolute values of their Walsh spectrum will be equivalent. As a result, any other property derived from the Walsh spectrum like nonlinearity will be affine invariant.

4.4 FDT_6

Equivalence classes under affine transformations have been studied in many works dating back to 1960s [14][15]. The number of equivalence classes for the whole set of $\mathcal{B}_n$ was counted up to $n = 6$. In 1991, Maiorana [22] discovered 150357 equivalence classes in $\mathcal{B}_6$. This classification was later confirmed by the works of Braeken et. al. [5], Fuller [13], and Langevin [17]. Based on Remark 4.1, we are ready to complete the function distribution table of dimension 6. We used the data, specifically the representative and the cardinality of each class from [17] to create FDT_6 (see Definition 3.2) by a new algorithm (see Algorithm 4.1).

In this algorithm, we mainly input the representatives and cardinalities of each of 150357 equivalence class, compute the frequency distributions of the Walsh coefficients and add up the cardinalities according to these frequency distributions.

Algorithm 4.1 Complete FDT_6 for ω <32

Input: $(f_1, c_1), (f_2, c_2), ..., (f_{150357}, c_{150357})$ ▷ Representatives and cardinalities of equivalence classes in $\mathcal{B}_6$

Output: FDT_6

$\omega \leftarrow 0$

for $s \leftarrow 0, 2^6$ **do** ▷ Initialize the table

while $\omega \leq 2^6$ **do**

$FDT_6[s][\omega] \leftarrow 0$

$\omega \leftarrow \omega + 2$

end while

end for

for $i \leftarrow 1, 150357$ **do** ▷ Main loop

$WalshSpectrum_i \leftarrow \text{ABS}(WHT(f_i))$

$(U, C) \leftarrow \text{CountUnique}(WalshSpectrum_i)$ ▷ see Algorithm B.1

for $j \leftarrow 1, length(U)$ **do**

if $U[j]$ <32 **then**

$FDT_6[C(j)][U(j)] \leftarrow FDT_6[C(j)][U(j)] + c_i$

end if

end for

end for

return FDT_6

The algorithm contains a CountUnique function, which is a procedure that gives unique elements and counts their occurrences from an input vector. This procedure has been given in Algortihm B.1.

In order to simplify numbers obtained by this algorithm, we define the normalized versions of $|\mathcal{S}(n,s,\omega)|$ and FDT_n to be as follows.

- $|\overline{\mathcal{S}(n,s,\omega)}| = \frac{|\mathcal{S}(n,s,\omega)|}{2^{n+1}}$
- $\overline{FDT_n}$ is the *normalized function distribution table* of $\mathcal{B}_n$ such that its (s,ω) entry contains $|\overline{\mathcal{S}(n,s,\omega)}|$.

Note that, these definitions are appropriate since $bx \oplus c$ part in the definition of affine equivalence contributes 2^{n+1} times for all functions in an equivalence class as $b \in \mathbb{F}_2^n$ and $c \in \mathbb{F}_2$. In other words, all $|\mathcal{S}(n,s,\omega)|$ values are divisible by 2^{n+1}.

The output of Algorithm 4.1 is also given as $\overline{FDT_6}$ in Appendix A. In order to ease the readability, the table is divided into four parts and shortened. One of the important values of this table is the one corresponding to $(s,\omega) = (64,8)$ entry, which is 42 386 176, the number of 6-variable bent functions modulo affine functions (i.e. $RM(1,6)$).

Proposition 3.8 Revisited

Using $\overline{FDT_6}$, the number of s values generated by Proposition 3.8 can be updated. It is now known that

$$\mathcal{S}_6 = \{s \mid 2 \leq s \leq 44 \text{ or } s \in \{46,48,51,54,56,60,63\}\}.$$

Using the proposition, a subset $\tilde{\mathcal{S}}_7$ of $\mathcal{S}_7$ can be constructed.

$$\{4,6,8,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,48,50,$$
$$52,54,56,58,60,62,64,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,$$
$$81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,$$
$$101,102,103,104,105,106,107,108,110,112,115,118,120,124,127\} \subset \mathcal{S}_7$$

Recall that, the numbers in this subset show that there exists Boolean functions having that many zeros in their Walsh spectra. Using this subset, one can proceed recursively and generate s values for higher dimensions. Table 4.2 shows the number of s values generated this way and the percentages taken with respect to 2^n, number of all possible s values. Compared to the cardinality of the set generated for $\mathcal{S}_7$ using $\mathcal{S}_5$ (see Table 3.2) , there is approximately $59\%(\frac{81-51}{51})$ increase in the number of generated s.

Table 4.2: Number of s values generated by Proposition 3.8 using $\mathcal{S}_6$

n	Number of all s (2^n)	#s generated	Percentage	$\lvert\mathcal{S}_n\rvert$
5	32	-	-	15
6	64	50	78.13% (50/64)	50
7	128	81	63.28% (81/128)	unknown
8	256	112	43.75% (112/256)	unknown
9	512	143	27.93% (143/512)	unknown
10	1024	174	16.99% (174/1024)	unknown

Number of Functions for each Nonlinearity in $\mathcal{B}_6$

Using the same data and a similar algorithm (Alg. B.2), a modified version of the Algorithm 4.1, we were able to obtain a survey on the number of functions for each level of nonlinearity, which was also given in [13] based on the Remark 4.1.

The output of Algorithm B.2 is shown as Table 4.3, which has been ordered from the highest nonlinearity level of 28 to the lowest 0 value. Note that, the counts below the nonlinearity 16 are equal to the values in $s = 1$ column of Table A.11. This is basically due to the fact that there is a single Walsh value in this part of FDT_n, which is maximal in the spectrum.

Table 4.3: Number of functions for each nonlinearity value

$nl(\mathbf{f})$	# f
28	5425430528
27	347227553792
26	1617838297055232
25	103868560519987200
24	1305039828998603264
23	3821934098435833856
22	5097726702198767616
21	4011570131804454912
20	2291582136636334080
19	1087405010755682304
18	458313050588725248
17	176395152249028608
16	62526600834171264
15	20418431982428160
14	6125529594728448
13	1681517927964672
12	420379481991168
11	95180260073472
10	19388571496448
9	3525194817536
8	566549167104
7	79515672576
6	9596719104
5	975937536
4	81328128
3	5332992
2	258048
1	8192
0	128

4.5 Local Connectivity

In 2003, Joanne Fuller constructed a new method of analyzing the effect of affine transformations with respect to their local connection neighborhood [13]. A summary of this approach is given below.

Definition 4.5. For a function $f(x) \in \mathcal{B}_n$, *the family of 1-local connection functions*, or simply *1-local neighborhood* is defined by the functions

$$f_\alpha(x) = \begin{cases} f(x), & x \neq \alpha \\ f(x) \oplus 1, & x = \alpha \end{cases}, \ \alpha = 0, 1, \cdots, 2^n - 1$$

This definition can be rephrased in terms of ANFs as follows.

Definition 4.6. For a function $f(x) \in \mathcal{B}_n$, define its (1-local) connection functions as

$$f_\alpha(x) = f(x) + \rho_\alpha(x),$$

where $\rho_\alpha(x) \in \{(x_0 + \alpha_0)(x_1 + \alpha_1) \cdots (x_n + \alpha_n) | \alpha = (\alpha_0, \alpha_1, \cdots, \alpha_n) \in \mathbb{F}_2^n\}$

It is clear that 1-local neighborhood of a function f consists of 2^n functions f_α, $\alpha \in \mathbb{F}_2^n$, such that $d(f, f_\alpha) = 1$. According to Definition 4.6, it is also seen that all connection functions of f have different algebraic degree than f, since ρ_α is always of degree n. As algebraic degree is an affine invariant property, different algebraic degrees imply non-equivalence of f with any of the f_α's. Thus, f_α's belong to different equivalence classes than f. In other terms, f_α's connect f to at least one and at most 2^n distinct equivalence classes. The foremost result of this structure is the following proposition.

Proposition 4.2. *Let f and g be two equivalent functions such that $g(x) = f(Ax \oplus a) \oplus bx \oplus c$ and let f_α be a 1-local connection function of f, then there exist a 1-local connection function g_β of g such that $g_\beta(x) = f_\alpha(Ax \oplus a) \oplus bx \oplus c$, $\beta = A^{-1}(\alpha \oplus a)$ and $\alpha = A\beta + a$.*

Proof.

$$f_\alpha(x) = \begin{cases} f(x), & x \neq \alpha \\ f(x) \oplus 1, & x = \alpha \end{cases} \Rightarrow$$

$$f_\alpha(Ax + a) \oplus bx \oplus c = \begin{cases} f(Ax \oplus a) \oplus bx \oplus c, & (Ax \oplus a) \neq \alpha \\ f(Ax \oplus a) \oplus bx \oplus c \oplus 1, & (Ax \oplus a) = \alpha \end{cases} \Rightarrow$$

$$f_\alpha(Ax + a) \oplus bx \oplus c = \begin{cases} g(x), & x \neq (A^{-1}(\alpha \oplus a)) \\ g(x) \oplus 1, & x = (A^{-1}(\alpha \oplus a)) \end{cases}$$

Without loss of generality, let $(A^{-1}(\alpha \oplus a)) = \beta$. Thus, $f_\alpha(Ax + a) \oplus bx \oplus c$ is equivalent to g_β such that $\beta = (A^{-1}(\alpha \oplus a))$ and $\alpha = A\beta + a$.

□

The main consequence of this result due to the non-singularity of A is the following corollary.

Corollary 4.3. *Let f and g be two equivalent Boolean functions. Then the 1-local neighborhood of f and g are composed of functions of same classes.*

4.6 A New Invariant Property

Corollary 4.3 establishes that the set of 2^n connection functions is invariant. Here, we propose another invariant based on this invariance.

Proposition 4.4. *Let f and g be two equivalent Boolean functions. Then the number of functions connecting f and g to any class in their 1-local neighborhood is the same.*

Proof. This is also a direct consequence of Proposition 4.2, since the proposition ensures that for each connection function f_α of f, there exists a connection function for g, say g_β, which is equivalent to f_α. f and g belong to the same class, while f_α and g_β belong to another class in 1-local neighborhood. □

The complement statement of this proposition can be employed to test two functions for equivalence.

Corollary 4.5. *For two Boolean functions f and g, if the counts of 1-local connection functions, which connect them to the same equivalence class, are different, f and g cannot be equivalent.*

Assume that the equivalence classes, where two arbitrary Boolean functions f and g reside, is connected to k and l distinct classes respectively. Let $L_f = \{r_1, r_2, \cdots, r_k\}$ and $L_g = \{t_1, t_2, \cdots, t_l\}$ be the lists of number of connections of f and g to their neighboring classes. Corollary 4.5 ensures that if $k \neq l$ or $L_f \not\equiv L_g$ then f and g cannot be equal.

4.7 Class Connection Digraphs

Fuller explains the importance of the local connectivity with the following words.

"The connectivity relationships that exist between equivalent functions provide a unique perspective of the inherent structure that exists between the classes. Given a single function from any class, we can effectively determine its position with respect to all other classes.".[13]

In the same work, in order to reflect this inherent structure, the local connectivity was represented visually by *class connection diagrams* based on the fact that any two equivalent functions will have the same connecting classes. In these

diagrams, each equivalence class is displayed as a single node and there is an edge between 'connected' classes. For Boolean functions of 4-variables the corresponding diagram is given in Figure 4.1.

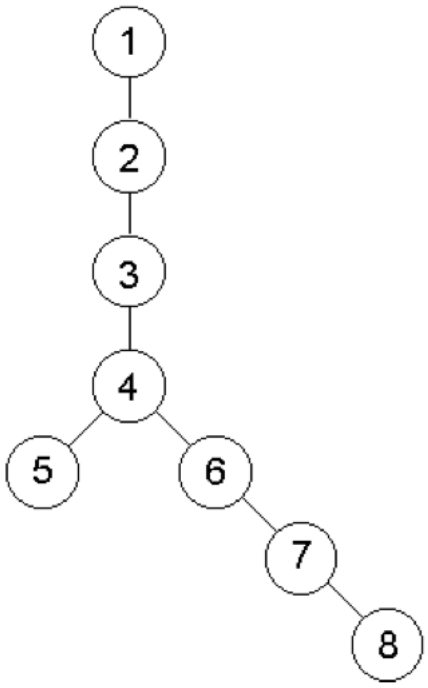

Figure 4.1: Class connection diagram for $n = 4$

We enrich the class connection diagram by adding the number of connection functions going in and out of each node (equivalence class), and thus convert the diagram to a *directed graph*, simply *digraph*.

Definition 4.7. *Class connection digraph* of $\mathcal{B}_n$ is a directed and weighted graph of equivalence classes in $\mathcal{B}_n$, where the number of connection functions are assigned to the directed edges.

Figure 4.2 and Figure 4.3 are the class connection digraphs of $\mathcal{B}_4$ and $\mathcal{B}_5$ respectively. Note that the sum of the weights directed out of each node adds up to 2^n, i.e. the number of total connecting functions. Moreover, the nodes are positioned according to the nonlinearity of the corresponding class, starting from 0,- the root node of affine class, and proceeding in ascending order. Thus, there are classes with maximal nonlinearity, e.g. bent classes for even n values, at the bottom of these digraphs.

In order to find the weights of the edges, we combine two invariant properties which allow us to distinguish between connection functions. These two are the distribution of absolute values of Walsh spectrum and the invariant proposed by

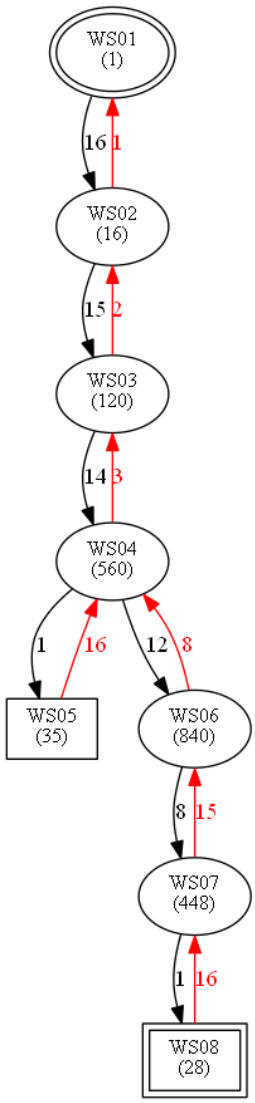

Figure 4.2: Class connection digraph for $\mathcal{B}_4$

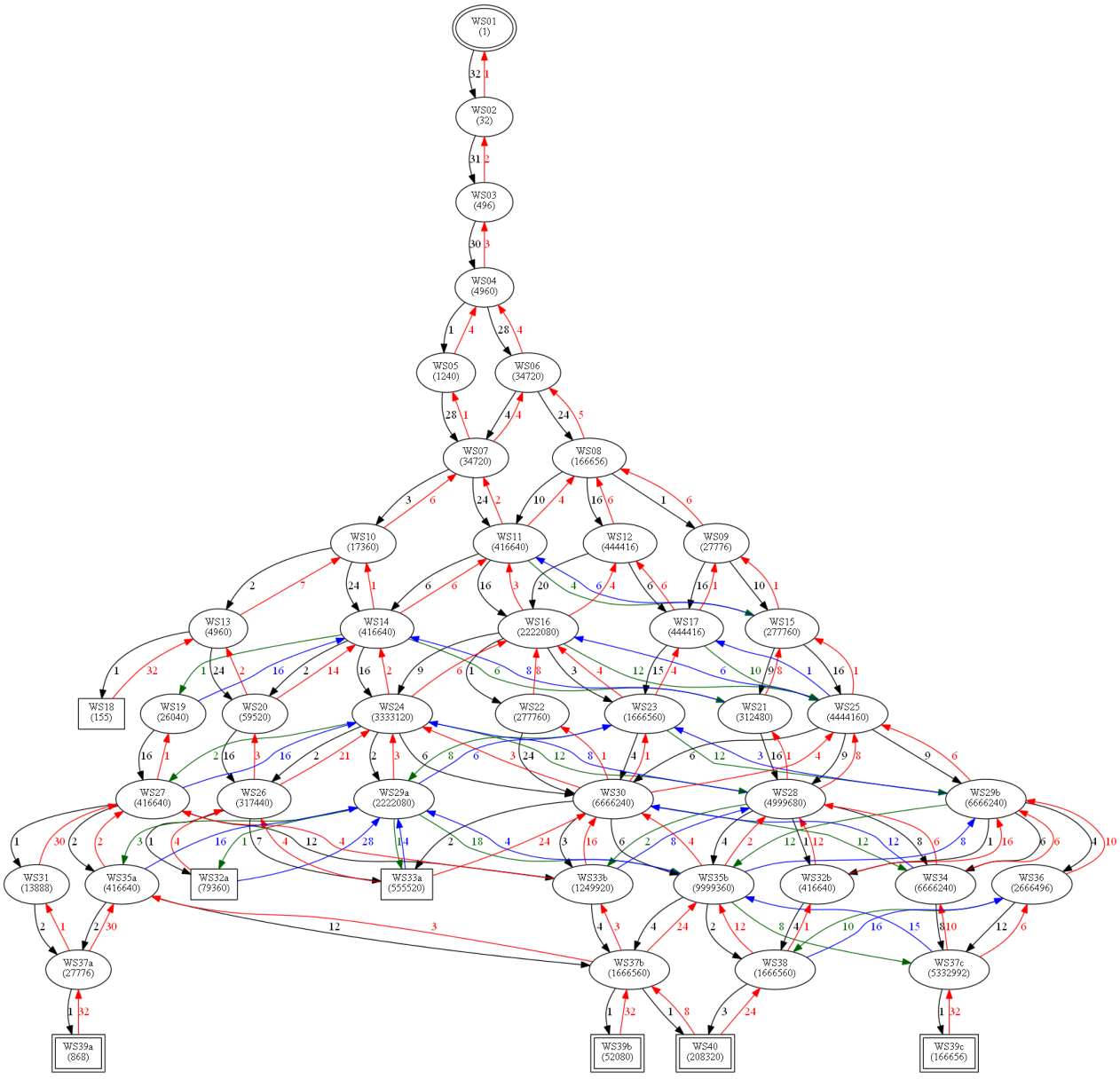

Figure 4.3: Class connection digraph for $\mathcal{B}_5$

Corollary 4.5. We compute a row of numbers by concatenating these two data that represent an equivalence class.

Recall from graph theory [1] that the *depth* of a node is the length of the path from the root to that node. The root node is at depth zero. The set of all nodes at a given depth is called a *level*.

Nodes in class connection digraphs are named with respect to the appearance of each new distinct distribution of absolute values of Walsh spectrum. For instance, WS05 represents the 5^{th} different Walsh spectrum distribution. On the other hand, small letter suffixes like a,b,c as in Figure 4.3 imply a different equivalence class with the same distribution. As can be seen in the figures, there are 8 and 40 distinct distributions for $\mathcal{B}_4$, $\mathcal{B}_5$ respectively. On the other hand, the numbers written in parenthesis inside each node correspond to the size of that equivalence class up to affine functions, i.e. mod $RM(1, n)$.

4.8 The Size of Equivalence Classes

In Section 4.4, FDT_6 is computed using the representatives and cardinalities of all equivalence classes in $\mathcal{B}_6$. It is in fact a challenging task to obtain these data for higher dimension even if it is not possible to list all classes even for $n = 7$ (see Table 4.1). However, fixing the algebraic degree or aiming at particular classes (e.g. bent functions) can be enough to be able to complete more cells in FDT_n. In this section, we focus on the calculation of class sizes for a given representative of that class.

Class Connection Digraphs induce an interesting and simple way to compute the size of equivalence classes. Before going into details about this computation, a brief review of the previous works is given below.

In previous works on equivalence classes [22][28], the size of each class is obtained as a byproduct during the classification phases. It is usually noted to be found by a computer program that applies affine transformations of variables until the equivalence classes are exhausted.

In [13], self mapping analysis was proposed to be able to count the number of functions in an equivalence class. An affine transformation that relates a Boolean function to itself is called a *self mapping* of f. Using this analysis, the following formula was found.

Proposition 4.6. *For $f \in B_n$ of equivalence class $\mathcal{C}(f)$ with θ self mappings, the number of functions in $\mathcal{C}(f)$ is computed by*

$$|\mathcal{C}(f)| = \frac{\mathcal{T}}{\theta},$$

where $\mathcal{T}$ is the total number of distinct affine transformations, i.e.

$$\mathcal{T} = 2^{2n+1} \prod_{i=0}^{n-1} (2^n - 2^i)$$

Since there is not any explicit formula for the number of self mappings, this formula still depends on the classification algorithms that produce this number. Though, the naive interval for the number of self mappings θ of a Boolean function f is given to be $1 \leq \theta \leq \frac{2^{2n+1} \prod_{i=0}^{n-1} (2^n - 2^i)}{2^{n+1}}$, derived simply from the trivial class sizes.

In [5], a formula for the number of functions in an equivalence class $\mathcal{C}(f)$ possessing a p-property is given.

$$\mathcal{N}_p = \mathcal{K}_{\mathcal{C}(f)} \sum_{f \in \mathcal{R}} B_f.$$

where B_f is the number of proper bases in the zero sets of f with respect to p-property, $\mathcal{R}$ is a representative coset and $\mathcal{K}_{\mathcal{C}(f)} = \frac{2n! |\mathcal{C}(f)|}{\prod_{i=0}^{n-1} (2^n - 2^i)}$.

The size of the class $\mathcal{C}(f)$ is a variable in this formula and to obtain it, the values $\mathcal{N}_p$ and B_f should be known beforehand, for which a direct computation is not known.

In conclusion, none of the previous works included a method for direct computation of the size of an equivalence classes. We will now explain a method for doing this.

In Figure 4.2 and Figure 4.3, it can be observed that the size of the classes written inside parenthesis for each node follows a combinatorial pattern. These numbers can be obtained by dividing the product of the incoming edge weights by the product of the outgoing edge weights on the path from the root node to the node of interest.

Proposition 4.7. *Let $\mathcal{C}_i$ and $\mathcal{C}_j$ be two nodes in a class connection digraph representing two equivalence classes having N_i and N_j number of functions. Let the weight of the edge directed from $\mathcal{C}_i$ to $\mathcal{C}_j$ be x and the weight of the opposite edge be y. Then,*

$$N_j = \frac{x}{y} N_i. \tag{4.2}$$

Proof. The proof of this proposition requires a combinatorial observation. Figure 4.4 shows two arbitrary nodes $\mathcal{C}_i$ and $\mathcal{C}_j$ connected as stated in the proposition.

The weight x represents the number of connection functions which transfer each function of class $\mathcal{C}_i$ to class $\mathcal{C}_j$. Thus, for each function in class $\mathcal{C}_i$, x functions are produced making a total of xN_i functions that belong to $\mathcal{C}_j$. However, these functions can include multiple instances of same functions. In other words, a group of functions from class $\mathcal{C}_i$ connect with same functions in class $\mathcal{C}_j$. Due

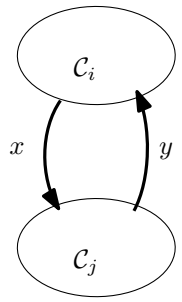

Figure 4.4: Two nodes from a class connection digraph

to the symmetry in 1-local neighborhood of each function of class $\mathcal{C}_i$ given by Corollary 4.3, $\frac{xN_i}{N_j}$ functions of class $\mathcal{C}_i$ meet at the same function in class $\mathcal{C}_j$.

Similarly, for each function in class $\mathcal{C}_j$, y functions are produced, which makes a total of yN_j functions. This implies that $\frac{yN_j}{N_i}$ functions of class $\mathcal{C}_j$ meet at the same function in class $\mathcal{C}_i$.

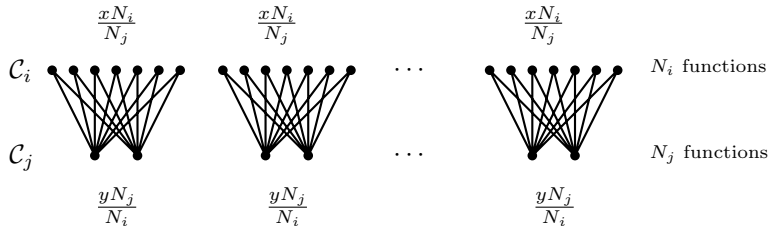

Figure 4.5: Structure between two connected classes

Thinking in reverse, these numbers mean that for each function in $\mathcal{C}_i$, there are $\frac{yN_j}{N_i}$ functions produced in $\mathcal{C}_j$ and for each function in $\mathcal{C}_j$, there are $\frac{xN_i}{N_j}$ functions produced in $\mathcal{C}_i$. As a result, group of $\frac{xN_i}{N_j}$ functions in $\mathcal{C}_i$ matches with a group of $\frac{yN_j}{N_i}$ functions in $\mathcal{C}_i$ (see Figure 4.5). Equating the number of groups, the following equality is found.

$$\frac{N_i}{\frac{xN_i}{N_j}} = \frac{N_j}{\frac{yN_j}{N_i}} \Rightarrow N_j = \frac{x}{y} N_i.$$

□

This proposition is better understood when seen as the conservation of total number of connection functions between two connecting classes as Equation 4.2 implies

$$y \cdot N_j = x \cdot N_i.$$

Corollary 4.8. *Let a Boolean function f with nonlinearity $nl(f)$ belong to an equivalence class $\mathcal{C}(f)$. Let $IW_f = \{i_1, i_2, \cdots, i_{nl(f)}\}$ and $OW_f = \{o_1, o_2, \cdots, o_{nl(f)}\}$ be the lists of weights of incoming and outgoing edges on a path of minimum length from the root node to the node of $\mathcal{C}(f)$. Then the size of the equivalence class of f is found by*

$$|\mathcal{C}(f)| = 2^{n+1} \prod_{j=1}^{nl(f)} \frac{i_j}{o_j}$$

Proof. This is a successive use of the Proposition 4.7 on a path, which starts with the well-known class of 2^{n+1} affine functions. □

Example 4.1. Assume that the number of bent functions in $\mathcal{B}_4$ is needed. There is only one equivalence class, say $\mathcal{C}_b(f)$, representing bent functions in this set and it is shown by 'WS08' in Figure 4.2. Let f be a representative bent function in this class. Then, $IW_f = \{16, 15, 14, 12, 8, 1\}$ and $OW_f = \{1, 2, 3, 8, 15, 16\}$. Thus,

$$|\mathcal{C}_b(f)| = 2^5 \cdot \frac{16 \cdot 15 \cdot 14 \cdot 12 \cdot 8 \cdot 1}{1 \cdot 2 \cdot 3 \cdot 8 \cdot 15 \cdot 16} = 896$$

Example 4.2. Let us find the number of functions in the class denoted by 'WS11' in Figure 4.3. It can be seen from the class connection digraph that there are three paths of minimum length that start with the root node and end at 'WS11'.

1. WS01 → WS02 → WS03 → WS04 → WS05 → WS07 → WS11
2. WS01 → WS02 → WS03 → WS04 → WS06 → WS07 → WS11
3. WS01 → WS02 → WS03 → WS04 → WS06 → WS08 → WS11

Listing the weights and applying the formula for each case, the followings are obtained.

1. $IW_f = \{32, 31, 30, 1, 28, 24\}$ and $OW_f = \{1, 2, 3, 4, 1, 2\}$.

$$64 \cdot \frac{32 \cdot 31 \cdot 30 \cdot 1 \cdot 28 \cdot 24}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 1 \cdot 2} = 64 \cdot 416640 = 26664960$$

2. $IW_f = \{32, 31, 30, 28, 4, 24\}$ and $OW_f = \{1, 2, 3, 4, 4, 2\}$.

$$64 \cdot \frac{32 \cdot 31 \cdot 30 \cdot 28 \cdot 4 \cdot 24}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 4 \cdot 2} = 64 \cdot 416640 = 26664960$$

3. $IW_f = \{32, 31, 30, 28, 24, 10\}$ and $OW_f = \{1, 2, 3, 4, 5, 4\}$.

$$64 \cdot \frac{32 \cdot 31 \cdot 30 \cdot 28 \cdot 24 \cdot 10}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 4} = 64 \cdot 416640 = 26664960$$

Remark 4.2. Instead of starting from the root node, the path in Corollary 4.8 can start from any class with known number of functions. Moreover, since the numbers in opposite directions would cancel out, it is not necessary to choose the shortest path, if one desires to find the number of functions as quick as possible.

Example 4.3. Assume that the number of functions in the equivalence class, represented by 'WS39a' in Figure 4.3 is given beforehand to be 868 (mod $RM(1,n)$) and the number of functions in the equivalence class 'WS39b' that has the same distribution of absolute values of Walsh spectrum is needed. Then, out of many paths connecting these two classes, two of them and the corresponding computation of the class size are as follows.

$$\text{WS39a} \to \text{WS37a} \to \text{WS35a} \to \text{WS37b} \to \text{WS39b (the shortest path)}$$

$$868 \cdot \frac{32 \cdot 30 \cdot 12 \cdot 1}{1 \cdot 2 \cdot 3 \cdot 32} = 52080$$

$$\text{WS39a} \to \text{WS37a} \to \text{WS31} \to \text{WS27} \to \text{WS33b} \to \text{WS37b}$$
$$\to \text{WS40} \to \text{WS38} \to \text{WS35b} \to \text{WS37b} \to \text{WS39b}$$

$$868 \cdot \frac{32 \cdot 1 \cdot 30 \cdot 12 \cdot 4 \cdot 1 \cdot 24 \cdot 12 \cdot 4 \cdot 1}{1 \cdot 2 \cdot 1 \cdot 4 \cdot 3 \cdot 8 \cdot 3 \cdot 2 \cdot 24 \cdot 32} = 52080$$

Recall that, a factor of 2^{n+1} should be kept in mind for the actual values of class sizes since the numbers in parenthesis are actual values divided by 2^{n+1}.

Remark 4.3. Note that the size of each class contained in the chosen paths are included in the computation.

Example 4.4. Take the shortest path in Example 4.3 for instance. The size of 'WS39a' is assumed to be known already.

i. Size of 'WS39b': $868 \cdot \frac{32 \cdot 30 \cdot 12 \cdot 1}{1 \cdot 2 \cdot 3 \cdot 32} = 52080$

ii. Size of 'WS37b': $868 \cdot \frac{32 \cdot 30 \cdot 12}{1 \cdot 2 \cdot 3} = 1666560$

iii. Size of 'WS35a': $868 \cdot \frac{32 \cdot 30}{1 \cdot 2} = 416640$

iv. Size of 'WS37a': $868 \cdot \frac{32}{1} = 27776$

4.9 Algorithm

Corollay 4.8 allows to derive an algorithm to find the number of functions in any equivalence class. The pseudocode of this algorithm is presented below. Unless otherwise stated, f refers to the truth table of a Boolean function f.

Algorithm 4.2 CountEquivalentFunctions(f)

Input: $f \in \mathcal{B}_n$

Output: Number of functions affine equivalent to f

$l \leftarrow$ FindClosestAffine(f) ▷ see Algorithm B.3

$c \leftarrow l \oplus f$ ▷ vector of bits to flip in f

while $c \neq \overline{0}$ **do** ▷ until all 1 bits in c has been flipped

$f_{temp} \leftarrow f$

$i \leftarrow$ find$(c, 1)$ ▷ locate first occurrence of 1 in c

$c[i] \leftarrow 0$ ▷ change this 1 to 0 in c

$f[i] \leftarrow f[i] \oplus 1$ ▷ making f closer to l in each loop

$CurrentRow \leftarrow$ ComputeRow(f_{temp})

$NextRow \leftarrow$ ComputeRow(f) ▷ see Algorithm B.4

$Weights_{in} \leftarrow Weights_{in} \cup \{$FindEdgeWeight$(f, CurrentRow)\}$

$Weights_{out} \leftarrow Weights_{out} \cup \{$FindEdgeWeight$(f_{temp}, NextRow)\}$ ▷ B.7

end while

$Product_{in} \leftarrow$ Product($Weights_{in}$)

$Product_{out} \leftarrow$ Product($Weights_{out}$) ▷ multiply all elements in the list

return $2^{n+1} \cdot Product_{in}/Product_{out}$

4.10 Complexity Analysis of the Algorithm

The structure of Algorithm 4.2 allows its complexity to be calculated easily. First of all, the main while loop is processed $nl(f)$ times. Since nonlinearity is maximum for bent functions, this main loop can repeat $2^{n-1} - 2^{n/2-1}$ times at most. Main loop calls only two sub-algorithms, ComputeRow (Alg. B.4) and FindEdgeWeight (Alg. B.7), both of which are called twice.

The first sub-algorithm ComputeRow contains a single FWT and GetAllEdge Weights functions. The most costly part is the GetAllEdgeWeights function. It contains 2^n FWT calls and 2^n CountConnectingWS, which contains 2^n FWT calls itself. Thus in total, ComputeRow costs $2^n\mathcal{O}(n2^n)+2^{2n}\mathcal{O}(n2^n) \approx \mathcal{O}(n2^{3n})$.

The second sub-algorithm called from the main loop is FindEdgeWeight. This function calls 2^n FWT and 2^n ComputeRow. This costs approximately $2^n\mathcal{O}(n2^{3n})$.

In total, CountEquivalentFunctions algorithm costs at most

$$(2^{n-1} - 2^{n/2-1}) \cdot 2 \cdot \left[\mathcal{O}(n2^{3n}) + 2^n\mathcal{O}(n2^{3n})\right] \approx \mathcal{O}(n2^{5n}),$$

which is basically 2^{4n} FWT calls.

Table 4.4: Complexity of Algortihm 4.2 with respect to n

n	$log_2(n2^{5n})$
1	5.00
2	11.00
3	16.59
4	22.00
5	27.32
6	32.59
7	37.81
8	43.00
9	48.17
10	53.32
11	58.46
12	63.59

Although the complexity of $\mathcal{O}(n2^{5n})$ does not allow the algorithm to be better than the exhaustive search for $n \neq 5$ when being used for complete classification, it stands as an improvement for higher n values. It takes $48 \cdot 2^{27.32} \approx 2^{32.91}$ and $150357 \cdot 2^{32.59} \approx 2^{49.79}$ complexities to verify all orbit sizes in $\mathcal{B}_5$ and $\mathcal{B}_6$, respectively. Still, listing all class cardinalities would not be feasible for $\mathcal{B}_7$, since it contains $63379147320777408548 \approx 2^{65.78}$ classes (see Table 4.1). It would take approximately $2^{103.59}$ to compute cardinality of each class even if the classes

were possible to store. However, given any function in $\mathcal{B}_7$, especially with good cryptographic properties, this algorithm is successful in determining the number of equivalent functions. Therefore, rather than using this algorithm for listing complete classifications, which will take too much time, it is preferable to utilize it to enumerate chosen classes of Boolean functions.

Moreover, since the theory allows to start from any class with known size, these complexities are just upper bounds implying the worst case scenario, which is starting from the affine class and going down to the bent classes. Thus, in best case scenario, there are two connection classes, one of whose size is known. In this case, the formula of Proposition 4.7 should be followed. Assuming that N_i is known, then x and y values can be found by just two FindEdgeWeight algorithm calls, which is approximately 2^{3n} FWT calls. In total, the minimum cost of finding a class size is $\mathcal{O}(n2^{4n})$

Table 4.5: Minimum cost of finding class sizes with respect to n

n	$log_2(n2^{4n})$
1	4.00
2	9.00
3	13.59
4	18.00
5	22.32
6	26.59
7	30.81
8	35.00
9	39.17
10	43.32
11	47.46
12	51.59

4.11 Conclusion

In this chapter, we used equivalence classes to improve results in Chapter 3. The notion and the importance of equivalence with respect to affine transformations, various facts and known affine invariant properties have been discussed. FDT_6 has been completed and given in Appendix A. A new affine invariant property that can be used for testing equivalence has been introduced. Using this property, a counting algorithm that gives the number of functions in an equivalence class has been created. There are several methods in literature that gives this number as a byproduct of classification algorithms. However, to the best of our knowledge, this is the first systematic algorithm that aims to find only this quantity.

As a final remark, each FDT_n in Appendix A can be verified by the above algorithm since a representative from each class is already provided. Moreover,

equivalence classes in $RM(3,7)/RM(2,7)$, whose representatives are given by Braeken et el. in [5] are also verified.

CHAPTER 5

CONCLUSION

5.1 Thesis Summary

Boolean functions are essential to many applications used in various scientific disciplines, most importantly in cryptography. Symmetric cryptographic systems in particular count on 'strong' Boolean functions as their core components. Hence, the design and analysis of Boolean functions is a very active area of study.

In this thesis, Boolean functions are studied with respect to their Walsh spectrum. Their existence, construction and enumeration have been investigated according to the distribution of values in Walsh spectrum.

The thesis is composed of five chapters. Chapter 1 provides the introduction to the theory of Boolean functions and the motivation of the thesis, while Chapter 2 gives preliminary technical information about the thesis. Chapter 3 starts with the main problem of interest, which asks the number of n-variable Boolean functions, whose Walsh spectrum contains a specified number s of a specified Walsh coefficient ω, exist in $\mathcal{B}_n$. Later in this chapter, a mathematical framework to study this problem is formed. The previous related results are adapted into this framework. Finally, new results based on several grounds like the effects of modifications on truth table or the Parseval's identity, are presented. Chapter 4 contains the improvement of the solution to the main problem by using the idea of equivalence classes. It starts with an introductory information on this idea and the invariant properties of equivalence classes. Distribution of Boolean functions of 6 variables according to the frequency of Walsh coefficients is completed with this preliminary information and the data obtained from [17]. In order to proceed with higher dimensions, a useful tool, the local connectivity proposed by [13], is also explained. Later, based on this theory, a new invariant propertyand a new algorithm to count the number of functions in an equivalence class are proposed.

5.2 Contributions of the Thesis

The research presented here has contributed to the existing literature about the theory of Boolean functions in several ways.

Firstly a new framework of parameters to give a better view for the distribution of Boolean functions with respect to the quantities of coefficients in their Walsh spectrum is proposed. Secondly, previous works on Walsh coefficients and nonlinearity is redefined in this framework so that they could be viewed from a different perspective. The highlights of the remaining contributions of the thesis are as follows.

I. Solution sets $\mathcal{S}(n, s, \omega)$ for all s and ω such that $n \leq 6$ are found. Thus, the distribution of Boolean functions with respect to the frequency of Walsh coefficients up to 6 variables are completed.

II. Exact values of $|\mathcal{S}(n, s, \omega)|$ for all s and n such that $\omega \geq 2^{n-1}$, which is in line with Wu's work in [42] are formulized. In other words, functions of arbitrary input length having Walsh coefficients larger than or equal to 2^{n-1} is completely determined.

III. Existence of n-variable Boolean functions with s many zeros in the Walsh spectrum for some s is provided by means of a concatenation type of construction method.

IV. Several other bounds and results are also found by exploiting the Parseval's Identity and other combinatorial observations.

V. A new invariant property that can be used to test the equivalence of Boolean functions based on local connectivity introduced by [13] is proposed.

VI. Finally, a first-of-its-kind algorithm to count the number of functions in any equivalence class using the proposed invariant property is presented.

5.3 Further Study

Classification of $\mathcal{B}_n$ for $n \leq 5$, can be easily accomplished by exhaustive search as the space is manageable. In order to complete the distribution of Boolean functions of 6 variables according to the frequency of Walsh coefficients, equivalence classes in $\mathcal{B}_6$ are used. However, for higher number of variables, even equivalence classes do not reduce the search space enough. Thus, either particular classes or specified algebraic degrees must be targeted. Although some results apply to functions of arbitrary variables, the distribution of 7-variable Boolean functions is still incomplete.

Existence of Boolean functions having s many zeros in their Walsh spectrum for a fair percentage of possible s values is determined in this thesis. However, It is still an open problem to determine the complete list of number of zeros that a Boolean function of more then 6 input variables may contain in its Walsh spectrum.

Affine invariant properties are also of great importance to the analysis of Boolean functions, since they are the main elements for distinguishing equivalence classes. The new invariant presented here provides a solid contribution to the literature,

mainly because it is used in a pioneering algorithm to count the number of functions in an equivalence class unlike any previous method. To be able to compute the cardinality of an equivalence class easily is also an essential means to enumerate Boolean function classes with 'good' cryptographic properties. Therefore, the algorithm devised in this thesis can be used to study those existing classes in literature. This algorithm can be employed to compute the size of any equivalence class feasibly up to $n = 11$ variables. In fact, an optimization of the algorithm, such as computing only a particular subset of the outgoing edge weights might also give the desired results. Therefore a further investigation can provide better results in this area.

REFERENCES

[1] L.W. Beineke and R.J. Wilson, eds. *Topics in Algebraic Graph Theory*. Vol. 102. Cambridge University Press, 2004.

[2] E.R. Berlekamp and L.R. Welch, *Weight Distributions of the Cosets of the (32, 6) Reed-Muller Code*, IEEE Transactions on Information Theory, 18(1), pp. 203–207, 1972.

[3] G. Boole, *An Investigation of The Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*, Originally published by Macmillan, London, 1854. Reprint by Dover, 1958.

[4] A. Braeken, *Cryptographic properties of Boolean functions and S-boxes*. Ph.D. Thesis, KU Leuven, Belgium, 2006.

[5] A. Braeken, Y. Borissov, S. Nikova, B. Preneel, *Classification of boolean functions of 6 variables or less with respect to some cryptographic properties*, Automata, Languages and Programming. Springer Berlin Heidelberg, pp. 324–334, 2005.

[6] A. Braeken, S. Nikova, Y. Borissov, *Classification of cubic Boolean functions in 7 variables*, Proc. of the 26th Symposium on Information Theory in the Benelux, Brussels, Belgium, 2005.

[7] C. Carlet, *Constructing balanced functions with optimum algebraic immunity*, IEEE International Symposium on Information Theory, pp. 451–455, 2007.

[8] C. Carlet, D. K. Dalai, K. C. Gupta, S. Maitra, *Algebraic immunity for cryptographically significant Boolean functions: Analysis and construction*, IEEE Transactions on Information Theory, 52 (7), pp. 3105–3121, 2006.

[9] C. Carlet, S. Mesnager, *On the supports of the Walsh transforms of Boolean functions*, IACR Cryptology ePrint Archive 256, 2004.

[10] C. Carlet, P. Sarkar, *Spectral domain analysis of correlation immune and resilient Boolean functions*, Finite Fields and Their Applications, 8 (1), pp. 120–130, 2002.

[11] N. Courtois and W. Meier, *Algebraic attacks on stream ciphers with linear feedback*, Eurocrypt 2003 (E. Biham, ed.), LNCS, vol. 2656, Springer-Verlag, pp. 34–359, 2003.

[12] J.F. Dillon, *Elementary Hadamard difference sets*, Ph.D. Thesis, University of Maryland, USA, 1974.

[13] J. Fuller, *Analysis of affine equivalent Boolean functions for cryptography*, Ph.D. Thesis, QUT, Australia, 2003.

[14] M.A. Harrison, *The Number of Transitivity Sets of Boolean Functions*, Journal of the Society for Industrial and Applied Mathematics 11, pp. 806–828, 1963.

[15] M.A. Harrison, *On the Classification of Boolean Functions by the General Linear and Affine Group*, Journal of the Society for Industrial and Applied Mathematics 12, pp. 284–299, 1964.

[16] X. D. Hou, *AGL (m, 2) acting on R (r, m)/R (s, m)*, Journal of Algebra 171.3, pp. 921–938, 1995.

[17] P. Langevin, *Classification of Boolean functions under the affine group*, `http://langevin.univ-tln.fr/project/agl/agl.html`, 2009.

[18] P. Langevin, G. Leander, *Counting all bent functions in dimension eight 99270589265934370305785861242880*, Designs, Codes and Cryptography, 59 (1), pp. 193–205, 2011.

[19] K. Limniotis, N. Kolokotronis, N. Kalouptsidis, *Constructing Boolean functions in odd number of variables with maximum algebraic immunity*, IEEE International Symposium on Information Theory Proceedings, pp. 2686–2690, 2011.

[20] O. A. Logachev, S. V. Smyshlyaev, V. V. Yashchenko, *On ρ-balanced Boolean functions*, Discrete Mathematics and Applications, 22 (3), pp. 345–352, 2012.

[21] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error Correcting Codes*, North- Holland, 1977.

[22] J. Maiorana, *A Classification of the Cosets of the Reed-Muller Code R(1,6).* Mathematics of Computation, 57 (195), pp. 403–414, 1991.

[23] S. Maitra, E. Pasalic, *Further constructions of resilient Boolean functions with very high nonlinearity*, IEEE Transactions on Information Theory, 48 (7), pp. 1825–1834, 2002.

[24] M. Matsui, *Linear cryptanalysis for DES cipher*, Eurocrypt 1993 (T. Helleseth, ed.), LNCS, vol. 950, Springer-Verlag, pp. 38–397, 1993.

[25] Q-S. Meng, H-G. Zhang, M. Yang, Z-Y. Wang, *Analysis of affinely equivalent Boolean functions*, Science in China Series F: Information Sciences 50 (3), pp.299–306, 2007.

[26] D. Pei, W. Qin, *The correlation of a Boolean function with its variables*, in: Proceedings of the First International Conference on Progress in Cryptology, INDOCRYPT '00, Springer-Verlag, pp. 1–8, 2000.

[27] P. Porwik, *Walsh coefficients distribution for some types of Boolean functions*, Institute of Computer Science, University of Silesia, 2004.

[28] B. Preneel, *Analysis and Design of Cryptographic Hash Functions*, Ph.D. Thesis, KU Leuven, Belgium, 1993.

[29] B. Preneel, R. Govaerts, J. Vandewalle, *Cryptographic properties of quadratic Boolean functions*, 1^{st} International Conference on Finite Fields and Applications, 1991.

[30] B. Preneel, O. Logachev, *Open problems in Boolean function theory: The cryptographer's view*, in: Boolean Functions in Cryptology and Information Security, NATO Science for Peace and Security Series, IOS Press Inc., pp. 343–351, 2008.

[31] B. Preneel, W.V. Leekwijck, L.V. Linden, R. Govaerts, and J. Vandewalle, *Propagation characteristics of Boolean functions*, In Advances in Cryptology-EUROCRYPT' 90, LNCS 437, pp. 55–165, Springer-Verlag, 1990.

[32] O. S. Rothaus, *On bent functions*, Journal of Combinatorial Theory, Series A, 20 (3), pp. 300–305, 1976.

[33] P. Sarkar and S. Maitra, *Construction of nonlinear Boolean functions with important cryptographic properties*, Advances in Cryptology, EUROCRYPT 2000, LNCS 1807, pp. 491–512, Springer-Verlag, Berlin/New York, 2000.

[34] J. L. Shanks, *Computation of the fast Walsh-Fourier transform*, IEEE Transactions on Computers, 100 (5), pp. 457–459,1969.

[35] C. E. Shannon, *A symbolic analysis of relay and switching circuits*, Electrical Engineering, 57 (12), pp. 713–723, 1938.

[36] T. Siegenthaler, *Correlation Immunity of Nonlinear Combining Functions for Cryptographic Applications*, IEEE Transactions on Information Theory, 30 (5), pp. 776–780, 1984.

[37] T. Siegenthaler, *Decrypting a class of stream ciphers using ciphertext only*, IEEE Transactions on Computers, C-34 (1), pp. 81–85, 1985.

[38] N.J.A. Sloane, Sequence A001289, *Number of equivalence classes of Boolean functions modulo linear functions*, The On-Line Encyclopedia of Integer Sequences, `http://oeis.org/A001289`, 2013.

[39] E. Uyan, Boolean function distributions with respect to Walsh coefficients `http://www.metu.edu.tr/~uerdener/phd/fdt.htm`, 2013.

[40] E. Uyan, Ç. Çalık, A. Doğanaksoy, *Counting Boolean functions with specified values in their Walsh spectrum*, Journal of Computational and Applied Mathematics, ISSN 0377-0427, `http://dx.doi.org/10.1016/j.cam.2013.06.035`, 2013.

[41] E. Uyan, A. Doğanaksoy, *Distribution of Boolean Functions of 6 Variables According to the Frequency of Walsh Coefficients*, 6^{th} International Information Security & Cryptology Conference, ISC Turkey 2013, Ankara, Turkey, 20-21 September 2013.

[42] C-K. Wu, *On distribution of Boolean functions with nonlinearity* $\leq 2^{n-2}$, Australasian J. Combin., 17, pp. 51–59, 1998.

APPENDIX A

Function Distribution Tables

Table A.1: FDT_1

$s \setminus \lvert\omega\rvert$	**0**	**2**
0	0	0
1	4	4
2	0	0

Table A.2: FDT_2

$s \setminus \lvert\omega\rvert$	**0**	**2**	**4**
0	8	8	8
1	0	0	8
2	0	0	0
3	8	0	0
4	0	8	0

Table A.3: FDT_3

$s \setminus \lvert\omega\rvert$	**0**	**2**	**4**	**6**	**8**
0	128	128	144	128	240
1	0	0	0	128	16
2	0	0	0	0	0
3	0	0	0	0	0
4	112	0	112	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	16	128	0	0	0
8	0	0	0	0	0

Table A.4: FDT_4

$s \setminus \lvert\omega\rvert$	**0**	**2**	**4**	**6**	**8**	**10**	**12**	**14**	**16**
0	33664	32768	33920	33280	37536	47616	61696	65024	65504
1	0	0	0	0	0	17920	3840	512	32
2	0	0	0	0	26880	0	0	0	0
3	0	0	0	17920	0	0	0	0	0
4	0	0	0	0	1120	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	26880	0	0	14336	0	0	0	0	0
7	0	0	3840	0	0	0	0	0	0
8	3840	0	26880	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	14336	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0
12	1120	17920	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
15	32	512	0	0	0	0	0	0	0
16	0	0	896	0	0	0	0	0	0

Table A.5: FDT_5, $|\omega| \leq 16$

$s \setminus \lvert\omega\rvert$	**0**	**2**	**4**	**6**	**8**	**10**	**12**	**14**	**16**
0	2181005312	2147483648	2181070848	2147803136	2186856192	2165897216	2351758336	2887755776	3647300416
1	0	0	0	0	0	59424768	483524608	1039933440	622182400
2	0	0	0	0	0	346644480	883752960	346644480	25474560
3	0	0	0	0	2222080	713287680	427750400	20633600	0
4	13332480	0	0	28887040	375531520	133324800	147292160	0	9920
5	0	0	0	0	0	426639360	0	0	0
6	170655744	0	0	26664960	739508224	449748992	888832	0	0
7	106659840	0	0	447272960	79360	0	0	0	0
8	666624000	0	0	0	774950400	0	0	0	0
9	284426240	0	0	142213120	0	0	0	0	0
10	475080704	0	888832	780394496	170655744	0	0	0	0
11	213319680	0	0	0	0	0	0	0	0
12	144435200	106659840	116659200	17776640	31109120	0	0	0	0
13	0	0	426639360	568852480	0	0	0	0	0
14	3809280	0	910417920	0	0	0	0	0	0
15	0	597295104	483556352	28442624	0	0	0	0	0
16	16086272	449748992	142213120	106659840	14054656	0	0	0	0
17	0	0	0	0	0	0	0	0	0
18	0	337756160	0	0	0	0	0	0	0
19	17776640	568852480	0	0	0	0	0	0	0
20	0	10665984	0	0	0	0	0	0	0
21	0	20316160	0	0	0	0	0	0	0
22	1666560	53329920	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0
24	79360	2539520	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0
28	9920	317440	31744000	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0
30	0	0	1777664	0	0	0	0	0	0
31	64	2048	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0

Table A.6: FDT_5, $|\omega| \geq 18$

s \ $\|\omega\|$	**18**	**20**	**22**	**24**	**26**	**28**	**30**	**32**
0	4079552512	4236971008	4282079232	4292665856	4294649856	4294935552	4294965248	4294967232
1	215414784	57996288	12888064	2301440	317440	31744	2048	64
2	0	0	0	0	0	0	0	0
...	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0

Table A.7: $\overline{FDT_6}$, $|\omega| \leq 6$

s \ $\|\omega\|$	**0**	**2**	**4**	**6**
0	72621222574527744	72057602467610624	72620544008126464	72057622083079424
1	0	0	0	0
2	105833226240	0	0	0
3	1722169774080	0	0	279982080
4	18251051857920	0	0	242534476800
5	87944051220480	0	0	72571355136
6	266917829165568	0	0	5049693798912
7	769143475666944	0	0	1041773322240
8	2138149299686400	0	0	46160060567040
9	4252057890877440	0	0	22294693048320
10	6105590049226752	3563462590464	5375655936	318935587815936
11	8202945889861632	0	0	344858407649280
12	11169686287323648	17618712330240	173133918720	1147171079476800
13	11950669988167680	134391398400	0	1634668654510080
14	9472893446016000	10751311872000	11087290368000	3155879210803200
15	6560061945163776	28670164992	260859303936	3799950102336192
16	4761952300046592	590073040908288	286056593956224	5966106700732416
17	3019781441617920	0	8090362183680	6417306866073600
18	1421987940696960	2137171158958080	2903602090072320	7885495224337920
19	487272272578560	2167009782497280	1202290648473600	6729634715811840
20	184216244655744	1362013145579520	1293492405637555 2	9404376692368896
21	117198498816000	6934460358131712	676423606118400	5145082754961408
22	24292389687552	5486797902458880	24611185665803520	6930253495388160
23	45663077376000	4270958641152000	1421993706577920	4389266748948480
24	128100272307840	15327122841354240	1962479576518 2720	3093122840801280
25	7570715443200	3923109128945664	1071088413954048	2437301570715648
26	99314340475392	8882293960802304	6217932249865728	1516734154801152
27	73120261559296	8271606505078784	250634358374400	724202892492800
28	17335115481600	2729334178099200	753267543104160	618725559029760
29	50189027696640	4319361943142400	7310892072960	145874343444480
30	29400528995712	3071917184716800	31322467233792	134730475364352
31	8641366917120	754841031081984	2016	19061180006400
32	16938738915840	1395015149316096	1310636113920	
33	1338081024000	117579781079040	0	2153622159360
34	2711626444800	218435229250560	0	2814379868160
35	151222321152	11670581035008	0	525278380032
36	466259115840	39555770268672	0	302853505024
37	5879623680	1633975418880	0	80634839040
38	142153901568	10712646346752	0	0
39	16297290240	1359966289920	0	0
40	16825173120	1207842693120	11293655826432	0
41	1959874560	138871111680	0	0
42	840133728	57568315392	84978881003520	5119672320
43	209986560	13439139840	0	0
44	142178400	9099417600	191545260318720	0
45	0	10665984	10695763427328	597295104
46	3749760	10543325184	160689405886464	0
47	0	0	25688915804160	0
48	241950660	15484842240	51895751792640	0
49	0	1142784000	16050995527680	0
50	0	0	6288285523968	0
51	17498880	1119928320	3226793472000	0
52	0	0	376225920000	0
53	0	0	179188531200	0
54	546840	34997760	34430018560	0
55	0	0	5711634432	0
56	11160	714240	281981952	0
57	0	0	279982080	0
58	0	0	104993280	0
59	0	0	0	0
60	651	41664	4999680	0
61	0	0	0	0
62	0	0	0	0
63	1	64	0	0
64	0	0	0	0

Table A.8: $\overline{FDT_6}$, $8 \leq |\omega| \leq 14$

$s \setminus \lvert\omega\rvert$	**8**	**10**	**12**	**14**
0	72622095976660224	72057689154119936	72625034638355584	72142467633545216
1	0	662473154560	24023638388736	747614794924032
2	0	9130356619776	207318618759168	2995695843194880
3	0	44785297890816	719428478963712	7242174646990080
4	2961685440	209499448052160	2188257251009760	11859982725648960
5	91554140160	581080576462848	4396261821235200	13938338817220608
6	5278134677760	1547109348830976	7844919431271552	12540151435773696
7	118695723612480	2917542785935680	10411747357832352	9616702651727040
8	686891401136640	5100761866346496	12066003006382080	6865856427064320
9	3207474708480	6941532029071360	11379448548235264	4128919532236800
10	209504767128384	8710898218998528	9359218987896960	1648147202994432
11	3289041327882240	8950279657439232	5771002952171520	356293099782144
12	13059723162336144	9129127308123648	4124139956982528	31220241776640
13	13074183198720	7456265936437248	1301540936048640	1612696780800
14	724262090509440	6917940343480320	962121643130880	22855680
15	9483549226012848	4811912521887744	312211255867392	10303340544
16	29265523912590336	3886013057246208	113765111566848	0
17	4959042600960	2320173419397120	4676820664320	0
18	302041308119040	1434637361332224	207628438929408	0
19	3691807869173760	674302121902080	53756559360	0
20	9531988828669440	296361927622656	107513118720	0
21	290808053760	87344169615360	84978881003520	0
22	15474924541440	27830050762752	5375655936	0
23	177256654848000	1451427102720	0	0
24	394568875975680	854505308160	11293655826432	0
25	385255342080	0	0	0
26	8648086487040	0	0	0
27	50166815784960	0	0	0
28	72125261493376	2712715264	0	0
29	684556185600	0	0	0
30	17263717801344	0	0	0
31	99358640640000	0	0	0
32	128254086213120	0	0	0
33	197107384320	0	0	0
34	7552796590080	0	0	0
35	45617816272896	0	0	0
36	53903749939200	0	0	0
37	24078458880	0	0	0
38	2019230760960	0	0	0
39	13923508838400	0	0	0
40	15514367016960	0	0	0
41	0	0	0	0
42	0	0	0	0
43	0	0	0	0
44	0	0	0	0
45	0	0	0	0
46	6089610240	0	0	0
47	100793548800	0	0	0
48	111866840064	0	0	0
49	0	0	0	0
...	0	0	0	0
63	0	0	0	0
64	42386176	0	0	0

Table A.9: $\overline{FDT_6}$, $16 \leq |\omega| \leq 22$

$s \setminus \lvert\omega\rvert$	**16**	**18**	**20**	**22**
0	73640137188883852	78376899729410752	91801043670119456	110117258751722048
1	5240629419250176	18947976160909824	29921358592984320	27393590186016768
2	12560023422051840	22835525169348864	17361660563856000	6108505058154240
3	17917219124686560	15589394844663360	4453518280108128	485968528731840
4	16964459254387680	6659042781484800	546544843474560	9837047888640
5	10864229316820992	1545493802065920	30350253459456	28222193664
6	4861454914971648	155468441216256	699893953920	281148672
7	1513645200518040	5332394704896	11945902080	0
8	418173220747800	53756559360	31997952	0
9	104723252838400	995491840	0	0
10	30139398955008	0	0	0
11	0	0	0	0
12	354120459840	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	241284036	0	0	0
17	0	0	0	0
...	0	0	0	0
64	0	0	0	0

Table A.10: $\overline{FDT_6}$, $24 \leq |\omega| \leq 30$

s \ $\|\omega\|$	**24**	**26**	**28**	**30**
0	125833798480977392	135582604969668032	140532277216848224	142737029667930944
1	16966291037375232	8345818667585280	3564137512349568	1376946651273984
2	1292022710254080	186233857770240	18768803969664	1211745943296
3	22914064812240	530580832320	4500024480	10707648
4	161782145280	0	42663936	0
5	0	0	0	0
6	291648	0	0	0
7	0	0	0	0
...	0	0	0	0
64	0	0	0	0

Table A.11: FDT_6, $|\omega|$>32 †

$\|\omega\|\backslash s$	**0**	**1**	**2**	**...**	2^n
34	18426325641727123456	20418431982428160	0	0	0
36	18440618544114823168	6125529594728448	0	0	0
38	18445062555781586944	1681517927964672	0	0	0
40	18446323694227560448	420379481991168	0	0	0
42	18446648893449478144	95180260073472	0	0	0
44	18446724685138055168	19388571496448	0	0	0
46	18446740548514734080	3525194817536	0	0	0
48	18446743507160384512	566549167104	0	0	0
50	18446743994193879040	79515672576	0	0	0
52	18446744064112832512	9596719104	0	0	0
54	18446744072733614080	975937536	0	0	0
56	18446744073628223488	81328128	0	0	0
58	18446744073704218624	5332992	0	0	0
60	18446744073709293568	258048	0	0	0
62	18446744073709543424	8192	0	0	0
64	18446744073709551488	128	0	0	0

† For readability, this is a transposed version of the conventional FDT_n seen in [**?**]

APPENDIX B

Algorithms

Algorithm B.1 CountUnique(Vector)

Input: $Vector$
Output: $(UniqueElementList, CountOfOccurrences)$
  $MaxVal \leftarrow \text{Max}(Vector)$
  for $i \leftarrow 0, MaxVal$ **do** ▷ Initialisation
    $counter[i] \leftarrow 0$
  end for
  for each *value* in Vector **do**
    $counter[value] \leftarrow counter[value] + 1$
  end for
  for $value \leftarrow 0, MaxVal$ **do**
    if $counter[value]>0$ **then**
      $UniqueElementList \leftarrow UniqueElementList \cup \{value\}$
      $CountOfOccurrences \leftarrow CountOfOccurrences \cup \{counter[value]\}$
    end if
  end for

Algorithm B.2 NumberofFunctionsPerNonlinearity()

Input: $(f_1, c_1), (f_2, c_2), ..., (f_{150357}, c_{150357})$ ▷ Representatives and cardinalities of equvalance classes in $\mathcal{B}_6$
Output: Number of functions for each nonlinearity in $\mathcal{B}_6$
  for $nl \leftarrow 0, 26$ **do** ▷ Initialisation
    $f_{counts}[nl] \leftarrow 0$
  end for
  for $i \leftarrow 1, 150357$ **do** ▷ Main loop
    $WalshSpectrum_i \leftarrow \text{Abs}(WHT(f_i))$
    $W_{max} \leftarrow \text{Max}(WalshSpectrum_i)*$
    $nl \leftarrow 2^{n-1} - \frac{W_{max}}{2}$
    $f_{counts}[nl] \leftarrow f_{counts}[nl] + c_i$
  end for
  return f_{counts}

Algorithm B.3 FindClosestAffine(f)

Input: $f \in \mathcal{B}_n$
Output: $l \in RM(1, n)$

```
WalshSpectrum ← Abs(WHT(f))
W_max ← Max(WalshSpectrum)
for index ← 1, length(WalshSpectrum) do
    if WalshSpectrum[index] = W_max then
        l ← index × G_RM(1,n)        ▷ using generator matrix of RM(1,n) code
        break                        ▷ first found affine will suffice
    end if
end for
return l
```

Algorithm B.4 ComputeRow(f)

Input: $f \in \mathcal{B}_n$
Output: row_f ▷ An array of data generated by f

```
WalshSpectrum ← Abs(WHT(f))
(U, C) ← CountUnique(WalshSpectrum)        ▷ see Algorithm B.1
A ← GetAllEdgeWeights(f)                   ▷ see Algorithm B.5
row_f ← [U || C || A]
return row_f
```

Algorithm B.5 GetAllEdgeWeights(f): Algorithm to find the weights of outgoing edges of f

Input: $f \in \mathcal{B}_n$
Output: All Weights ▷ weights of outgoing edges of f

```
ConnectionFunctions ← 0        ▷ initialize matrix of connection functions
for j ← 0, 2^n do
    f[j] ← f[j] ⊕ 1
    WalshSpectrum ← Abs(WHT(f))
    (U, C) ← CountUnique(WalshSpectrum)        ▷ see Algorithm B.1
    WSfrequencies ← CountConnectionWS(f)       ▷ see Algorithm B.6
    CurrentRow ← [U || C || WSfrequencies]
    ConnectingFunctions(j) ← CurrentRow
    f[j] ← f[j] ⊕ 1
end for
(U, C) ← CountUnique(ConnectionFunctions)
weights ← C
return weights
```

Algorithm B.6 CountConnectionWS(f): Algorithm to find the number of occurence of each distinct Walsh Spectra of Connecting Functions of f

Input: $f \in \mathcal{B}_n$
Output: $WSCounts$ ▷ distribution of Walsh spectra of connection functions

```
ConnectionFunctionsWS ← 0                          ▷ initialize matrix of Walsh spectra
for j ← 0, 2^n do
    f[j] ← f[j] ⊕ 1
    WalshSpectrum ← Abs(WHT(f))
    (U, C) ← CountUnique(WalshSpectrum)            ▷ see Algorithm B.1
    CurrentRow ← [U || C]
    ConnectingFunctionsWS(j) ← CurrentRow
    f[j] ← f[j] ⊕ 1
end for
(UniqueWS, WSCounts) ← CountUnique(ConnectionFunctionsWS)
return WSCounts
```

Algorithm B.7 FindEdgeWeight(f,row_g): Algorithm to find the number of connection functions between two neighbor functions f to g

Input: $f \in \mathcal{B}_n$
Output: $weight$ ▷ weight of the edge between given classes

```
ConnectionFunctionRows ← 0
for j ← 0, 2^n do
    f[j] ← f[j] ⊕ 1
    newrow ← ComputeRow(f)                         ▷ see Algorithm B.4
    ConnectionFunctionRows(j) ← newrow
    f[j] ← f[j] ⊕ 1
end for
weight ← count(ConnectionFunctionRows, row_g)      ▷ count row_g among all
return weight
```

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Uyan, Erdener
Nationality: Turkish
Date and Place of Birth: 1985, Edirne
Email: uerdener@metu.edu.tr

EDUCATION

Degree	**Institution**	**Year of Graduation**
Ph.D. on BS.	Department of Cryptography, METU	2013
B.S.	Department of Mathematics, METU	2008
High School	Robert College	2004

AWARDS AND HONORS

Award	**Institution**	**Year**
Academic Performance Award	Middle East Technical University	2011
Honor Roll	Middle East Technical University	2005,2007,2008
Dora Aksoy Award	Robert College	2004

PROFESSIONAL EXPERIENCE

Year	**Place**	**Enrollment**
2009-2013	Middle East Technical University	Research Assistant
2005-2013	Middle East Technical University	System Administrator

PUBLICATIONS

E. Uyan, A. Doğanaksoy, *Distribution of Boolean Functions of 6 Variables According to the Frequency of Walsh Coefficients*, 6^{th} International Information Security & Cryptology Conference, ISC Turkey 2013, Ankara, Turkey, 20-21 September 2013.

E. Uyan, Ç. Çalık, A. Doğanaksoy, *Counting Boolean functions with specified values in their Walsh spectrum*, Journal of Computational and Applied Mathematics, ISSN 0377-0427, http://dx.doi.org/10.1016/j.cam.2013.06.035, 1 July 2013.

M. S. Turan, E. Uyan, *Near-Collisions for the Reduced Round Versions of Some Second SHA-3 Compression Functions using Hill Climbing*, Springer-Verlag Berlin Heidelberg 2010, LNCS 6498, G.Gong and K.C. Gupta Eds.: INDOCRYPT 2010, Haydarabad, India, pp. 131-143, 12-15 December 2010.

M. S. Turan, E. Uyan, *Practical Near-Collisions for Reduced Round Blake, Fugue, Hamsi and JH*, Information Technology Laboratory Publications, NIST, USA, 23 August 2010.

B. Bilgin, N. Öztop, E. Uyan, *A Survey on Rebound Attack*, 4^{th} International Information Security & Cryptology Conference, Proceedings, ISC Turkey 2010, Ankara, Turkey, pp. 242-246, 6-8 May 2010.