BOOLEAN FUNCTIONS WITH EXCELLENT CRYPTOGRAPHIC PROPERTIES IN AUTOCORRELATION AND WALSH SPECTRA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SELÇUK KAVUT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2008

Approval of the thesis:

BOOLEAN FUNCTIONS WITH EXCELLENT CRYPTOGRAPHIC PROPERTIES IN AUTOCORRELATION AND WALSH SPECTRA

submitted by **SELÇUK KAVUT** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmet Erkmen ________________
Head of Department, **Electrical and Electronics Engineering**

Assoc. Prof. Dr. Melek Diker Yücel ________________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Yalçın Tanık
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Melek Diker Yücel
Electrical and Electronics Engineering Dept., METU ________________

Assoc. Prof. Dr. Ali Doğanaksoy
Mathematics Department, METU ________________

Prof. Dr. Ferruh Özbudak
Mathematics Department, METU ________________

Assist. Prof. Dr. Ali Aydın Selçuk
Computer Engineering Dept., Bilkent 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: SELÇUK KAVUT

Signature :

ABSTRACT

BOOLEAN FUNCTIONS WITH EXCELLENT CRYPTOGRAPHIC PROPERTIES
IN AUTOCORRELATION AND WALSH SPECTRA

Kavut, Selçuk
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Assoc. Prof. Dr. Melek Diker Yücel

August 2008, 78 pages

We introduce a steepest-descent-like search algorithm for the design of Boolean functions, yielding multiple desirable cryptographic properties in their Walsh and autocorrelation spectra together. The algorithm finds some Boolean functions on 9, 10, 11, 13 variables with very good cryptographic properties unattained in the literature. More specifically, we have discovered 9-variable rotation symmetric Boolean functions (RSBFs) having nonlinearity of 241, which exceeds the bent concatenation bound and has remained as an open question in the literature for almost three decades. We have then shown that there is no RSBF having nonlinearity greater than 241, and that there are 8×189 many RSBFs having nonlinearity of 241, such that, among them there are only two that are different up to the affine equivalence. We also propose a generalization to RSBFs and dihedral symmetric Boolean functions (DSBFs), which improves the nonlinearity result of 9-variable Boolean functions to 242. Further, we classify all possible permutations ($9! = 362,880$) on the input variables of 9-variable Boolean functions and find that there are only 30 classes, which are different with respect to the linear equivalence of invariant Boolean functions under some permutations. Some of these classes and their subsets yield new 9-variable Boolean functions having the nonlinearity of 242 with different autocorrelation spectra from those of the Boolean functions found in generalized RSBF and DSBF classes. Moreover, we have attained 13-variable balanced Boolean functions having nonlinearity of 4036 which is greater than the bent concatenation bound of 4032, and improves the recent result of 4034.

Furthermore, we have found 10-variable Boolean functions having first order resiliency and a nonlinearity of 492, which was posed as an open question at Crypto 2000. We have also discovered balanced Boolean functions on n =9, 10, 11 variables having absolute indicator value less than $2^{\lceil \frac{n}{2} \rceil}$. Earlier the existence of such functions were known for 15 and 21 variables.

Keywords: Autocorrelation, Boolean Function, Cryptography, Heuristic Search, Nonlinearity.

ÖZ

ÖZİLİNTİ VE WALSH SPEKTRUMLARINDA ÜSTÜN KRİPTOGRAFİK
ÖZELLİKLERE SAHİP BOOLE İŞLEVLERİ

Kavut, Selçuk
Doktora, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Doç. Dr. Melek Diker Yücel

Ağustos 2008, 78 sayfa

Walsh ve özilinti spektrumlarında çeşitli arzulanan kriptografik özelliklere sahip Boole işlevleri tasarımı için en dik iniş prensibine dayalı arama algoritması geliştirilmiştir. Algoritma çok iyi kriptografik özelliklere sahip, literatürde daha önce elde edilememiş bazı 9, 10, 11, 13 değişkenli Boole işlevleri ortaya çıkarmıştır. Özellikle, literatürde yaklaşık olarak otuz senedir açık bir soru olarak bulunan, doğrusal olmama ölçütü 241 olan 9 değişkenli döngüsel simetrik Boole işlevleri (DSBİ) elde edilmiştir. Doğrusal olmama ölçütü 241'den büyük DSBİ olmadığı gösterilmiş ve doğrusal olmama ölçütü 241 olan DSBİ'lerin sayısının 8×189 adet olduğu bulunarak, bunların içinden sadece iki tanesinin ilgin eşdeğerliğe göre farklı olduğu ortaya çıkarılmıştır. Sonrasında, döngüsel simetrik ve ikidüzlemli simetrik Boole işlevleri (İSBİ) genelleştirilmiş ve bunun sonucunda 9 değişkenli Boole işlevleri için başarılan doğrusal olmama ölçütü 242'ye çıkarılmıştır. Bununla birlikte, 9 değişkenli Boole işlevlerinin giriş değişkenlerine uygalanabilir bütün devrişimler ($9! = 362,880$) sınıflandırılmış ve devrişime göre değişimsiz Boole işlevlerinin doğrusal eşdeğerliği bakımından sadece 30 sınıfın farklı olduğu bulunmuştur. Bu sınıfların bazılarında ve bunların altkümelerinde, genelleştirilmiş DSBİ ve İSBİ sınıflarınlarında elde edilen 242 doğrusal olmama ölçütüne sahip 9 değişkenli Boole işlevlerinin özilinti spektrumlarından farklı özilinti spektrumları bulunan yeni 242 doğrusal olmama ölçütüne sahip 9 değişkenli Boole işlevleri ortaya çıkarılmıştır. Bunun yanısıra, en yüksek doğrusal olmama ölçütüne sahip çift değişkenli iki işlevin birbirine bağlanması ile bulunan dengeli işlevler için doğrusal olmama sınırı 4032'den büyük, doğrusal olmama ölçütü 4036 olan 13 değişkenli dengeli Boole işlevleri üretilmiştir. Bu sonuç, yakın zamanda elde edilen 4034 doğrusal olmama değerini geliştirmiştir.

Ayrıca, Crypto 2000'de açık soru olarak ortaya konulan birinci derece esnekliğe sahip ve doğrusal olmama ölçütü 492 olan 10 değişkenli Boole işlevleri ortaya çıkarılmıştır. Bundan başka, mutlak gösterge değeri $2^{\lceil \frac{n}{2} \rceil}$'den küçük n =9, 10, 11 değişkenli dengeli Boole işlevleri elde edilmiştir. Bu tür Boole işlevleri daha önce 15 ve 21 değişkenliler için bilinmekteydi.

Anahtar Kelimeler: Boole İşlevi, Doğrusal Olmama, Kriptografi, Özilinti, Buluşsal Arama.

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to Assoc. Prof. Dr. Melek Diker Yücel for her guidance, patience, advice, criticism, encouragements and insight throughout this thesis. I have been extremely fortunate to receive her supervision. It is a great pleasure for me to thank Dr. Subhamoy Maitra for his academic collaboration and continuous motivation.

I acknowledge Dr. Gregor Leander for his discussion on linear equivalence of Boolean functions that are invariant under some permutations, which helped us to find one missing class in Table 5.3.

I am very pleased to express my sincere appreciation to Ferit Üzer and Gökhan Muzaffer Güvensen for having such friendly environment over the last two years.

Last but not the least, I am grateful to my dear family for their understanding, faith, and support.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background on Cryptography

Cryptography is the art and science of protecting and hiding a message from those who do not possess the *key* used to obscure that message. The security of a cryptographic system is usually based on some secret key, which is usually a binary sequence of sufficient length. The message to be protected (unencrypted data) is referred to as *plaintext*, while its obscured form (encrypted data) is called *ciphertext*. Plaintext and ciphertext are considered as the sequences of characters from an alphabet. In practice the alphabet consists of binary digits (bits). The process of hiding the plaintext within the ciphertext is termed *encryption* and the reverse process of extracting information from ciphertext is called *decryption*. A *cipher* is a cryptographic algorithm, which performs encryption and decryption.

In his vital paper [63] for modern cryptography, Shannon presented the principles of *confusion* and *diffusion*. Confusion obscures the relationship between the elements of plaintext and the elements of ciphertext, while diffusion spreads the influence of the plaintext elements over the ciphertext elements. Both principles try o achieve the same goal: hiding the statistical features of plaintext. *Cryptanalysis* is the study of recovering the information, hidden in the ciphertext, without access to the key. The science that encompasses both cryptography and cryptanalysis is referred to as *cryptology*.

There are mainly two classes of cryptosystems: *asymmetric* (*public-key*) and *symmetric* (*secret-key*). In asymmetric cryptography [12], each user has two distinct but mathematically related keys: a *public key* that is available to everyone and a *private key* that is kept secret from everybody. In order to send a secret message to a user, one encrypts the plaintext with the public key of that user. Decryption is possible only with the related private key uniquely owned by the intended user. Design principles of the cryptosystem make the calculation of the private key from the public key computationally infeasible.

Symmetric cryptographic algorithms have a single secret key, shared by the sender and the receiver, to perform both encryption and decryption. Symmetric cryptographic algorithms can be divided into *block ciphers* and *stream ciphers*. A block cipher operates on large blocks of data, while a stream cipher encrypts the plaintext symbols (usually bits) one at a time by combining with the keystream. For a given key, the transformation used by the block cipher is fixed, whereas that used by the stream cipher varies depending on cipher's state. Block ciphers can be used to construct other cryptographic primitives such as hash functions, message authentication codes and pseudorandom number generators. To fulfill the principles of confusion and diffusion, both block and stream ciphers make use of Boolean functions having desirable cryptographic properties.

1.2 Boolean Functions in Cryptography

Boolean functions, having either multi-output (also called *vector-output*) or single-output, constitute crucial components in secret-key cryptosystems. In fact, a standard block cipher (such as Data Encryption Standard (DES) [48] or its successor Advanced Encryption Standard (AES) [49]) can be considered as a vector-output Boolean function depending on the key. However, it is infeasible to analyze these functions due to the large number of input variables. On the other hand, vector Boolean functions on small number of variables such as substitution boxes (S-boxes) are iteratively used in block ciphers. Substitution boxes apply a nonlinear transformation on their input, hence their characteristics have significant effects on the strength of the entire system. Highly nonlinear S-boxes are required since in linear cryptanalysis [41], the linear combinations of the component functions of an S-box are approximated by linear Boolean functions. On the other hand, in differential cryptanalysis [3], the autocorrelation properties of S-boxes such as Strict Avalanche Criteria (SAC) [70] and Propagation Characteristics (PC) [54] are exploited; which makes Boolean functions with high values of autocorrelation undesirable.

In the design of a block cipher, the involved single-output Boolean functions must separately satisfy the desired cryptographic properties since an $n \times m$ vector Boolean function that maps n bits to m bits is composed of m many, n-variable single-output functions, each mapping n bits to a single bit. In case of stream ciphers, single-output n-variable Boolean functions are used as nonlinear combiners of Linear Feedback Shift Registers (LFSRs) to introduce nonlinearity [57]. In Figure 1.1, the combination generator is shown as a classic stream cipher model.

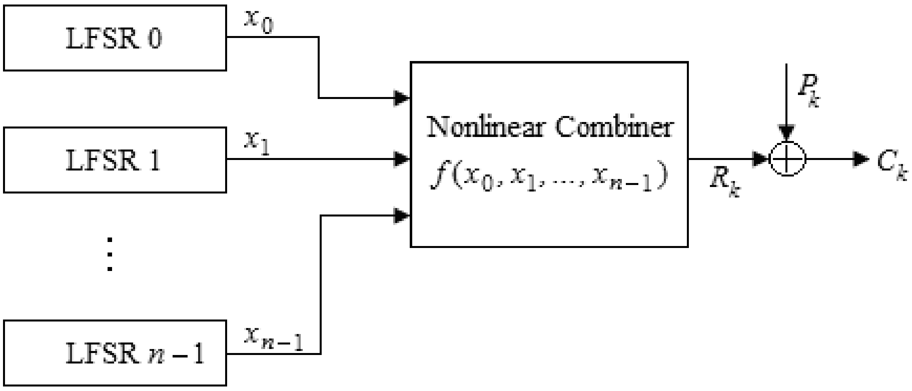

Figure 1.1: Combination generator

The stream $\{R_k\}$ is called the *keystream*, which is modulo 2 added to the plaintext stream $\{P_k\}$ to produce the ciphertext stream $\{C_k\}$. As the internal structure of n LFSRs and the Boolean function f can be public, the security of this cryptosystem depends on the initial state of the registers, which constitutes the secret key. To make the cryptosystem secure against cryptanalysis, the keystream R_k is desired to appear random. In fact, if it were truly random then the resulting cryptosystem, known as one time pad or Vernam cipher [69], would provide *unconditional security*.

The underlying single-output, n-variable Boolean function f must satisfy several cryptographic properties such as balancedness, high algebraic degree, high nonlinearity and correlation immunity to resist against cryptanalytic attacks available in symmetric key cryptography literature. Balancedness requires an equal number of 0's and 1's in the truth table of f to prevent the keystream from having statistical bias. High algebraic degree of f is a necessary condition to provide high *linear complexity* [13] of the keystream. The function f with low nonlinearity makes the cryptosystem vulnerable to Best Affine Approximation (BAA) attack [13] in which the keystream is approximated by an affine Boolean function. Thus f is desired to possess high nonlinearity. In general, it is easier to mount a cryptanalytic attack using linear approximations including a small number of input bits than using linear approximations involving a large number of input bits of f. This leads to the notion of correlation immunity and resiliency [13, 64, 65, 18], which are among the desired properties of cryptographically strong functions.

In this thesis we concentrate on single-output Boolean functions. Henceforth, we will use the term “Boolean functions” while referring to such functions. Boolean functions are also important in error correcting codes. For instance, Reed-Muller and Kerdock codes can be defined as sets of Boolean functions. In this thesis, we obtain important results on the covering radius of the first order Reed-Muller codes having codewords of length $2^9 = 512$, which corresponds to the maximum possible nonlinearity achieved by 9-variable Boolean functions.

1.3 Contributions of the Thesis

As discussed in the previous section, Boolean functions with high nonlinearity and low autocorrelation are important building blocks in cryptographic applications. For odd number of input variables n, the problem of constructing Boolean functions with very high nonlinearity is also related to the upper bound $\lfloor 2^{n-1} - 2^{\frac{n}{2}-1} \rfloor$ on the covering radius of the first order Reed-Muller code [21], which is later improved [23] as $2\lfloor 2^{n-2} - 2^{\frac{n}{2}-2} \rfloor$. Until 2006, the maximum nonlinearity known for the Boolean functions on $n = 9, 11, 13$ variables was the *bent concatenation bound* of $(2^{n-1} - 2^{\frac{n-1}{2}})$. The bound is simply achieved by the concatenation of two bent functions on $(m = n - 1)$ variables, where bent functions are Boolean functions on even number of variables m attaining maximum nonlinearity of $(2^{m-1} - 2^{\frac{m}{2}-1})$. Our four main contributions are related to these concepts. Firstly, in 2006, we discovered [25] 9-variable Boolean Functions with nonlinearity 241 $(= 2^{9-1} - 2^{\frac{9-1}{2}} + 1)$ in the class of Rotation Symmetric Boolean Functions (RSBFs), which led to the construction of functions with nonlinearity exceeding the bent concatenation bound by $1 \times 2^{\frac{n-9}{2}}$, for odd $n \geq 9$. Such functions were attained utilizing the steepest-descent-like iterative algorithm that first appeared in [29] and then suitably modified in [25] for a search in the class of RSBFs.

Considering a Boolean function f as a mapping from $GF(2^n) \rightarrow GF(2)$, the functions for which $f(\alpha^2) = f(\alpha)$ for any $\alpha \in GF(2^n)$, are referred to as idempotents [15, 16]. In [52], 15-variable Patterson-Wiedemann functions having nonlinearity 16276 $(= 2^{15-1} - 2^{\frac{15-1}{2}} + 20)$ are identified in the idempotent class. As pointed out in [15, 16], the idempotents can be seen as RSBFs with a proper choice of basis. Motivated by this, we studied the RSBF class and discovered 9-variable Boolean functions having nonlinearity 241. On the other hand, we exploited the nice combinatorial structure of the Walsh spectra for RSBFs on odd number of variables [43] to carry out the exhaustive search of 9-variable RSBFs having nonlinearity > 240, with considerable computational effort. Consequently, we found [24] that there are 8×189 many RSBFs having nonlinearity 241, which we showed as being the maximum possible value in this class. Further, utilizing some variants of binary nonsingular circulant matrices, we showed [24] that there are only two different 9-variable RSBFs having nonlinearity 241 up to the affine equivalence.

In 2007, as the second important contribution of this study, we proposed [30] the generalized k-RSBFs, as functions that satisfy $f(\alpha^{2^k}) = f(\alpha)$, where $1 \leq k \mid n$. Note that if $k = 1$, the resulting functions are the same as idempotents; whereas for $k = n$ the entire space of n-variable Boolean functions is covered. In the space of generalized k-RSBFs, imposing the condition of invariance under the action of dihedral group, we have defined the class of generalized k-DSBFs as a subset of k-RSBFs. Then, we have used the steepest-descent-like iterative algorithm in [25] for a search in the generalized 3-DSBF and 3-RSBF classes. As our third main contribution, this search successfully ended up [30] with 9-variable functions in both of these classes, achieving nonlinearity 242. This result shows that the covering radius of the first order Reed-Muller code $R(1, 9)$ is at least 242. This result is also important for $n = 11$ and $n = 13$, since the bent concatenation of 9-variable functions with nonlinearity 242 leads to the construction of 11-variable and 13-variable functions with nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 2 \times 2^{\frac{n-9}{2}})$, which exceeds the bent concatenation bound by $2 \times 2^{\frac{n-9}{2}}$. However, we should mention that for odd $n \geq 15$, the nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^{\frac{n-15}{2}})$ given in [52] that can be obtained by concatenating 15-variable Patterson-Wiedemann functions is still greater than the nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 2 \times 2^{\frac{n-9}{2}})$.

Further, in this thesis, knowing the fact that RSBFs, DSBFs, as well as the generalized k-RSBFs and k-DSBFs are invariant under some special types of permutations on input vectors, we have investigated [27] on the same search problem from a different direction and considered the possibility of other '*rich*' classes that are invariant under some permutations. Linearly equivalent Boolean functions have the same nonlinearity; therefore, while searching for highly nonlinear functions, it is quite logical to classify all $n!$ permutations up to the linear equivalence of Boolean functions that are invariant under them. More specifically, for 9-variable Boolean functions, we have classified $9!$ many permutations into 30 classes which are different up to the linear equivalence of Boolean functions that are invariant under them. Then for each class, by picking up a representative permutation arbitrarily, we have searched the corresponding set of Boolean functions. Consequently, in some of these sets, we have obtained [27] 9-variable Boolean functions with nonlinearity 242. So, our aim of defining other '*rich*' classes has been accomplished, as the fourth main contribution of this study. However, the functions mentioned so far do not contain any zero in their Walsh spectra, and hence, they cannot be linearly transformed to balanced functions.

In [34], Maitra used the 9-variable Boolean functions with nonlinearity 242, which we presented in [30], to construct a 13-variable balanced function having nonlinearity $(2^{13-1} - 2^{\frac{13-1}{2}} + 2 = 4034)$. That was the first demonstration of balanced Boolean functions on odd number of variables having nonlinearity strictly greater than the bent concatenation bound for number of input variables less than 15. We modified the search algorithm used in [34], and improved Maitra's result by arriving at 13-variable balanced functions having nonlinearity 4036 [36]. We consider this result as one of the side contributions of our study.

In 1995, it was conjectured [72] that for any balanced function on odd number of input variables n, the maximum absolute value in the autocorrelation spectrum is greater than or equal to $2^{\frac{n+1}{2}}$. Since then, the conjecture has been disproved by modifying 15-variable functions with nonlinearity 16276 given in [52] for $n = 15$ [39] and $n = 21$ [17]. As for another side contribution of our study, we have demonstrated [25] functions disproving the conjecture for odd $n < 15$. Our systematic search in the RSBF class shows that there exist balanced functions on 9 and 11 variables having maximum absolute value less than $2^{\frac{n+1}{2}}$.

In [60], a tight upper bound on nonlinearity has been proposed for resilient Boolean functions and the existence of some Boolean functions on 7 to 10 variables satisfying the bound has been posed as an open problem. Since then, the construction of these functions has been a challenging question. Some of them are presented in [51, 37, 68, 58]. As for another side contribution of our study, we found [25] a 10-variable 1-resilient function having nonlinearity 492, which was one of the unattained functions listed in [60]; previously, the best achieved nonlinearity of 10-variable 1-resilient functions was 488 [37].

1.4 Outline of the Thesis

After giving some preliminary material, related to this thesis, on Boolean functions in Chapter 2, we introduce our steepest-descent-like iterative search strategy and present its results attained in the class of Rotation Symmetric Boolean Functions (RSBFs) in Chapter 3. In particular, an RSBF on 9 variables having nonlinearity 241 is presented along with some other important Boolean functions having very good cryptographic properties in their Walsh and autocorrelation spectra together. The material of Chapter 3 are based on [25].

In Chapter 4, 9-variable RSBFs having nonlinearity > 240 are enumerated by an efficient exhaustive search strategy. It is found that there is no RSBF having nonlinearity > 241 and there are 8×189 many RSBFs having nonlinearity 241. Further it is proved that there are only two different 9-variable RSBFs having nonlinearity 241 up to the affine equivalence. This chapter is based on [24].

In Chapter 5, we improve the nonlinearity result of 241 by suitably generalizing the classes of RSBFs and Dihedral Symmetric Boolean Functions (DSBFs) and present several 9-variable Boolean functions having nonlinearity of 242. Then, we classify all possible permutations on input variables of 9-variable Boolean functions with respect to the linear equivalence of Boolean functions that are invariant under some permutations, which yields new 9-variable Boolean functions having nonlinearity 242 with different autocorrelation spectra from those of the functions found in generalized RSBF and DSBF classes. For this chapter the material are obtained from [30, 27].

In Chapter 6, we have attained 13-variable balanced functions having nonlinearity 4036 which is greater than the bent concatenation bound of 4032, and improves the recent result [34] of 4034. The material in this chapter are based on [36].

Finally, Chapter 7 is devoted to the conclusions, which provides a summary of our work and related open problems.

CHAPTER 2

PRELIMINARIES

An n-variable Boolean function $f(x)$ produces a single-bit result for each n-bit input vector $x = (x_0, \ldots, x_{n-1})$, which may be considered as a mapping from $\{0,1\}^n$ into $\{0,1\}$. $f(x)$ is basically represented by its *truth table*, that is, a binary vector of length 2^n,

$$f = [f(0,0,\ldots,0), f(1,0,\ldots,0), f(0,1,\ldots,0), \ldots, f(1,1,\ldots,1)]. \tag{2.1}$$

We represent the set of all n-variable Boolean functions by $\mathcal{B}_n$; clearly $|\mathcal{B}_n| = 2^{2^n}$. A binary vector g has the *Hamming weight* $wt(g)$ equal to the number of its nonzero elements. The *Hamming distance* between two binary vectors g and h, both having the same length, is defined as the number of places for which g and h differ, i.e., $d(g,h) = wt(g \oplus h)$, where $\oplus$ denotes the addition over $GF(2)$. An n-variable Boolean function f is called *balanced* if $wt(f) = 2^{n-1}$.

Algebraic Normal Form and Degree. The *algebraic normal form* (ANF) of a Boolean function $f(x)$ is defined as its unique representation in the form of a multivariate polynomial over $GF(2)$,

$$f(x_0,\ldots,x_{n-1}) = c \oplus \bigoplus_{0 \le i \le n-1} a_i x_i \oplus \bigoplus_{0 \le i < j \le n-1} a_{ij} x_i x_j \oplus \ldots \oplus a_{01\ldots n-1} x_0 x_1 \ldots x_{n-1}, \tag{2.2}$$

where the coefficients $c, a_i, a_{ij}, \ldots, a_{01\ldots n-1} \in \{0,1\}$. The *algebraic degree*, or simply the degree of f, is the number of variables in the highest order product term with nonzero coefficient, which is denoted by $deg(f)$.

Affine and Linear Boolean Functions. A Boolean function $f(x)$ having degree at most one is called an *affine* function of $x = (x_0, \ldots, x_{n-1}) \in \{0,1\}^n$. Its ANF is given by

$$f(x) = w_0x_0 \oplus w_1x_1 \oplus \ldots \oplus w_{n-1}x_{n-1} \oplus c =< w, x > \oplus c, \tag{2.3}$$

where $c \in \{0, 1\}$, $w = (w_0, \ldots, w_{n-1}) \in \{0, 1\}^n$, and $< w, x >$ represents the inner product of w and x. An affine function with the constant term $c = 0$ is called *linear*. The set of all n-variable affine (respectively linear) functions is denoted by A_n (respectively L_n).

Walsh Hadamard Transform. For a Boolean function f the Walsh Hadamard transform is a real valued function over $\{0, 1\}^n$ which is defined as

$$W_f(w) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus <x,w>}. \tag{2.4}$$

We refer to the vector $W_f = [W_f(0, 0, \cdots, 0)$, $W_f(1, 0, \cdots, 0)$, $W_f(0, 1, \cdots, 0)$, ..., $W_f(1, 1, \cdots, 1)]$ as the *Walsh spectrum* , or simply the spectrum of the function f. The Boolean functions f and g are said to have *nonintersecting Walsh spectra* [60, Lemma 7] if and only if $W_f(w) \neq 0 \Rightarrow W_g(w) = 0$ and $W_g(w) \neq 0 \Rightarrow W_f(w) = 0$ for all $w \in \{0, 1\}^n$.

Nonlinearity. The nonlinearity of an n-variable Boolean function f is defined as its minimum distance to any affine function, i.e.,

$$nl(f) = \min_{g \in A_n}(d(f, g)). \tag{2.5}$$

In terms of Walsh spectrum, the nonlinearity of f is given by

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{w \in \{0,1\}^n} |W_f(w)|. \tag{2.6}$$

Boolean functions used in cryptographic systems must be highly nonlinear to resist Best Affine Approximation (BAA) and correlation attacks [6, 13].

Correlation Immunity and Resiliency. Zhen and Massey [18] have provided a spectral characterization of correlation immune functions, which we use as the definition here. A Boolean function f is m-th order correlation immune (respectively m-resilient) if and only if its Walsh transform satisfies

$$W_f(w) = 0, \text{ for } 1 \le wt(w) \le m \text{ (respectively } 0 \le wt(w) \le m). \tag{2.7}$$

Parseval's Theorem. It states that for an n-variable Boolean function f, the sum of squared Walsh spectrum is constant and equal to 2^{2n}:

$$\sum_{w \in \{0,1\}^n} (W_f(w))^2 = 2^{2n}. \tag{2.8}$$

Autocorrelation Function. The autocorrelation function of a Boolean function f is given by

$$r_f(d) = \sum_{x \in \{0,1\}^n} (-1)^{f(x) \oplus f(x \oplus d)}, \tag{2.9}$$

where $d \in \{0,1\}^n$. The autocorrelation value having maximum magnitude (excluding the value at the origin which is equal to 2^n) is also known as the absolute indicator [72] and denoted as:

$$\Delta_f = \max_{d \in \{0,1\}^n, d \neq (0,\ldots,0)} |r_f(d)|. \tag{2.10}$$

Propagation Characteristics (PC) and Strict Avalanche Criteria (SAC) [70, 54] are important properties of Boolean functions to be used in S-boxes. A function is said to satisfy PC(k), if

$$r_f(d) = 0 \; for \; 1 \le wt(d) \le k. \tag{2.11}$$

An n-variable Boolean function (n is even) is called *bent* if the Walsh spectrum is flat, i.e., $W_f(w) = 2^{\frac{n}{2}}$ for all $w \in \{0,1\}^n$. Bent functions achieves the maximum nonlinearity of $(2^{n-1} - 2^{\frac{n}{2}-1})$ and absolute indicator value of 0 (i.e., $r_f(d) = 0$ for all $d \neq 0, \in \{0,1\}^n$). These functions exist only when n is even. In [71], the squared spectral distances from bent autocorrelation and bent Walsh spectra are related by the following theorem.

Theorem 2.1. $\sum_{d \neq 0} r_f{}^2(d) = 2^{-n} \sum_w (W_f{}^2(w) - 2^n)^2$.

Proof. Let us denote the Walsh Hadamard transform of $r_f(d)$ by $R_f(w)$. The following fact, which states that the autocorrelation and squared Walsh spectra form a transform pair, is used to prove the theorem.

$$\begin{aligned}
R_f(w) &= \sum_d r_f(d)(-1)^{<w,d>}, \\
&= \sum_d \sum_x (-1)^{f(x)}(-1)^{f(x\oplus d)}(-1)^{<w,d>}, \\
&= \sum_x \sum_y (-1)^{f(x)}(-1)^{f(y)}(-1)^{<w,x\oplus y>}, \\
&= \sum_x (-1)^{f(x)}(-1)^{<w,x>} \sum_y (-1)^{f(y)}(-1)^{<w,y>}, \\
&= W_f{}^2(w). \qquad (2.12)
\end{aligned}$$

Then, using this result together with Parseval's theorem,

$$\begin{aligned}
\sum_{d\neq 0} r_f{}^2(d) &= \sum_d r_f{}^2(d) - r_f{}^2(0), \\
&= \sum_d r_f{}^2(d) - 2^{2n}, \\
&= \sum_d 2^{-n} \sum_w R_f(w)(-1)^{<x,w>} 2^{-n} \sum_v R_f(v)(-1)^{<x,v>} - 2^{2n}, \\
&= 2^{-2n} \sum_w R_f(w) \sum_v R_f(v) \sum_d (-1)^{<x,w\oplus v>} - 2^{2n}, \\
&= 2^{-2n} \sum_w R_f(w) \sum_v R_f(v) 2^n \delta(w\oplus v) - 2^{2n}, \\
&= 2^{-n} \sum_w R_f{}^2(w) - 2^{2n}, \\
&= 2^{-n} \sum_w W_f{}^4(w) - 2^{2n}, \\
&= 2^{-n}(\sum_w W_f{}^4(w) - 2^{n+1} \sum_w W_f{}^2(w) + 2^{3n}), \\
&= 2^{-n} \sum_w (W_f{}^2(w) - 2^n)^2, \qquad (2.13)
\end{aligned}$$

where $\delta(w \oplus v)$ is the Kronecker delta function which is nonzero only when $w \oplus v = 0$.

In our steepest-descent-like search algorithm, we mostly use $\sum_{d\neq 0} r_f{}^2(d)$ as the cost function, which minimizes the squared distance to bent functions both in terms of Walsh and autocorrelation spectra, as can be seen from Theorem 1.

A Boolean function is balanced if and only if its Walsh spectrum value is zero at the origin. On the other hand, if an unbalanced Boolean function $g(x)$ contains a zero in its Walsh spectrum except the origin, say $W_g(u) = 0$ and $u \neq (0, \ldots, 0)$, it can be linearly transformed into a balanced function $f(x) = g(x) \oplus < x, u >$, which has the same nonlinearity and absolute indicator; i.e., $nl(f) = nl(g)$ and $\Delta_f = \Delta_g$.

Following the notation used in [60], we define the profile of a Boolean function by (n, m, d, σ) as its (input variable length, resiliency order, degree, nonlinearity). Adding the last entry Δ to the notation, by an $(n, m, d, \sigma, \Delta)$ function we denote an n-variable, m-resilient function with degree d, nonlinearity σ and absolute indicator Δ. By $(n, -1, d, \sigma, \Delta)$ we mean unbalanced functions and by $(n, 0, d, \sigma, \Delta)$ we mean balanced functions.

2.1 Group Action by Permutation Groups

A group G is said to act on a set X if there is a mapping $\phi : G \times X \to X$ denoted as $g \cdot x$, which satisfies the following two axioms for all elements $x \in X$.

1. $e \cdot x = x$ where e stands for the identity element of G.

2. $g \cdot (h \cdot x) = (gh) \cdot x$ for all $g, h \in G$.

The mapping ϕ is called the *group action* and the set X is called a G-set. The orbit of x is defined as the set $G(x) = \{g \cdot x \mid g \in G\}$, i.e., the group action moves x to its orbit. As the set of orbits of X under the action of G, denoted by $\mathcal{G}$, constitutes a partition of X, the corresponding equivalence relation is defined by $x \sim y$ iff there exists a $g \in G$ such that $g \cdot x = y$. Hence, the orbits form the equivalence classes under this relation.

Let G be a permutation group acting on $\{0, 1\}^n$, and consider the class of n-variable Boolean functions which are invariant under the action of G, i.e., any Boolean function f in the class satisfies the condition for each $x \in \{0, 1\}^n$, $f(x) = f(y)$, for all $y \in G(x)$. As a consequence of the invariance property, the class composes a subclass of $\mathcal{B}_n$, and knowing the number of orbits, i.e., $|\mathcal{G}|$, it contains $2^{|\mathcal{G}|}$ many n-variable Boolean functions, each satisfying the given condition. The value of $|\mathcal{G}|$ can be determined using Burnside's Lemma.

Lemma 1 (Burnside's Lemma). Let G be a group of permutations acting on a set X, and $fix_X(g) = \{x \in X \mid g \cdot x = x\}$ for each $g \in G$. Then the number of orbits induced on X is given by $\frac{1}{|G|}\sum_{g \in G} |fix_X(g)|$.

Let us represent an orbit by its lexicographically the first element, and denote the representative element as Λ_i. The Boolean function f which is invariant under the action of G, can be represented by $(f(\Lambda_0), \ldots, f(\Lambda_{|\mathcal{G}|-1}))$, where $\Lambda_0, \ldots, \Lambda_{|\mathcal{G}|-1}$ are again arranged lexicographically. Clearly, this representation is shorter than the truth table of f. Further, it can be shown [68] that $W_f(u) = W_f(v)$ if $u \in G(v)$, implying that the Walsh spectrum of f can be at most $|\mathcal{G}|$ valued. Then, defining a $|\mathcal{G}| \times |\mathcal{G}|$ matrix $\mathcal{A}$ as $\mathcal{A}_{i,j} = \sum_{x \in G_{\Lambda_i}} (-1)^{<x,\Lambda_j>}$ [68], the Walsh spectrum of f can be calculated as

$$W_f(\Lambda_j) = \sum_{i=0}^{|\mathcal{G}|-1} (-1)^{f(\Lambda_i)} \mathcal{A}_{i,j}. \tag{2.14}$$

In [43], a nice structure in the matrix $\mathcal{A}$ is obtained by applying some permutations on the representative elements $\Lambda_0, \Lambda_1, \ldots, \Lambda_{|\mathcal{G}|-1}$ for Boolean functions on odd number of input variables n. Let $\hat{\Lambda}_i$ denote the complement of Λ_i, then, for odd n, there is a one-to-one correspondence between the orbits of even weight Λ_i's and the orbits of odd weight Λ_i's by $\Lambda_i \to \hat{\Lambda}_i$. So, the set of orbits can be divided into two subsets (of same cardinality) containing representative elements of even weights and odd weights.

Now, consider the ordering of the Λ_i's as follows. Let us permute Λ_i's so that the first $\frac{|\mathcal{G}|}{2}$ representative elements correspond to the orbits of even weights in lexicographical order and the second $\frac{|\mathcal{G}|}{2}$ representative elements correspond to the complements of the first $\frac{|\mathcal{G}|}{2}$ orbits in the same order. That means the i-th ($i = 0, \ldots, \frac{|\mathcal{G}|}{2} - 1$) orbit representative Λ_i, where Λ_i is the i-th element in the new order, corresponds to the orbit representative $\hat{\Lambda}_{\frac{|\mathcal{G}|}{2}+i}$. Consequently, the permuted matrix, denoted by $\mathcal{A}^\pi$, has the form [43]

$$\mathcal{A}^\pi = \left[\begin{array}{c|c} \mathcal{H} & \mathcal{H} \\ \hline \mathcal{H} & -\mathcal{H} \end{array}\right]. \tag{2.15}$$

where $\mathcal{H}$ is a sub matrix of $\mathcal{A}^\pi$.

2.2 Rotation Symmetric Boolean Functions

Letting $(x_0, x_1, \ldots, x_{n-1}) \in \{0,1\}^n$, the (left) k-cyclic shift operator ${\rho^k}_n$ on n-tuples is defined as

$$
{\rho^k}_n(x_0, x_1, \ldots, x_{n-1}) = (x_{(0+k) \bmod n}, \ldots, x_{(n-1+k) \bmod n}), \tag{2.16}
$$

for $1 \leq k \leq n$. A Boolean function f is called *rotation symmetric* if for each input $(x_0, \ldots, x_{n-1}) \in \{0,1\}^n$, $f({\rho^1}_n(x_0, \ldots, x_{n-1})) = f(x_0, \ldots, x_{n-1})$. That is, RSBFs are invariant under all cyclic rotations of the inputs. The inputs of a rotation symmetric Boolean function can be divided into *orbits* so that each orbit consists of all cyclic shifts of one input. An orbit generated by $(x_0, x_1, \ldots, x_{n-1})$ is denoted by $G_n(x_0, x_1, \ldots, x_{n-1}) = \{{\rho^k}_n(x_0, x_1, \ldots, x_{n-1}) \mid 1 \leq k \leq n\}$ and the number of such orbits is g_n ($\approx 2^{\frac{2^n}{n}}$). More specifically, g_n is equal to $\frac{1}{n}\sum_{t|n} \phi(t) 2^{\frac{n}{t}}$ [67], where $\phi(t)$ is the Euler's phi-function. The total number of n-variable RSBFs is 2^{g_n}.

CHAPTER 3

SEARCH IN ROTATION SYMMETRIC CLASS

3.1 Introduction

On odd number of input variables n, constructing Boolean functions with maximum possible nonlinearity is an unsettled open problem in the area of symmetric cryptography and combinatorics. The problem is also related to the upper bound $\lfloor 2^{n-1} - 2^{\frac{n}{2}-1} \rfloor$ on the covering radius of the first order Reed-Muller code [21], which is later improved [23] as $2\lfloor 2^{n-2} - 2^{\frac{n}{2}-2} \rfloor$. Boolean functions on even number of input variables n, attaining maximum nonlinearity of $(2^{n-1} - 2^{\frac{n}{2}-1})$ are called the bent functions [56]. For odd number of input variables n, the nonlinearity value $(2^{n-1} - 2^{\frac{n-1}{2}})$ is known as the *bent concatenation bound*, since the concatenation of two bent functions on $(n-1)$ variables yields n-variable Boolean functions achieving this bound.

For odd $n \leq 7$, it is known that the maximum nonlinearity is equal to the bent concatenation bound [2, 47]. Clearly, as the number of n-variable Boolean functions (2^{2^n}) increases super-exponentially as n increases, exhaustive search of the whole space is not feasible for $n \geq 7$ with currently available hardware. Therefore, for any search attempt, different subclasses of Boolean functions are always significant and interesting.

In 1983, Patterson and Wiedemann [52] demonstrated a construction in the idempotent class, of 15-variable Boolean functions with nonlinearity 16276 (exceeding the bent concatenation bound by 20), using combinatorial techniques and search methods. Since then, it has been possible to get functions with nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^{\frac{n-15}{2}})$ for odd $n \geq 15$, which exceeds the bent concatenation bound by $20 \times 2^{\frac{n-15}{2}}$. Until 2006, the maximum nonlinearity known for the cases of $n = 9, 11, 13$ was equal to the bent concatenation bound. In 2006, 9-variable Rotation Symmetric Boolean Functions (RSBFs) with nonlinearity 241 $(= 2^{9-1} - 2^{\frac{9-1}{2}} + 1)$ were discovered [25], which led to the construction of functions with nonlinearity exceeding the bent concatenation bound by $1 \times 2^{\frac{n-9}{2}}$, for odd $n \geq 9$. Such functions were attained utilizing the steepest-descent-like iterative algorithm that first appeared in [29] and then suitably modified in [25] for a search in the class of RSBFs.

In symmetric cryptography, highly nonlinear Boolean functions having low absolute indicator value (Δ_f) are desired to satisfy confusion and diffusion criteria. In 1995, it was conjectured [72] that the maximum absolute value $\Delta_f \geq 2^{\frac{n+1}{2}}$ in the autocorrelation spectrum of any balanced Boolean function on odd number of input variables n. Since then, the conjecture has been disproved by modifying 15-variable Boolean functions with nonlinearity 16276 given in [52] for $n = 15$ [39] and $n = 21$ [17]. For the first time, we have demonstrated such functions for odd $n < 15$. Our systematic search in the RSBF class shows that there exist balanced Boolean functions on 9, 10 and 11 variables having $\Delta_f < 2^{\lceil \frac{n}{2} \rceil}$. For even number of input variables n, 8-variable balanced Boolean functions having $\Delta_f = 2^{\frac{n}{2}}$ are attained experimentally in [9, 28, 29], yielding better absolute indicator value than that of the construction proposed in [35] for which $\Delta_f \leq 2^{\frac{n}{2}} + \Delta_g$, where f and g are balanced functions on n and $\frac{n}{2}$ variables respectively. So far there was no evidence of balanced Boolean functions on even number of input variables n having $\Delta_f < 2^{\frac{n}{2}}$.

In particular, we have attained 9-variable (11-variable) RSBFs having nonlinearity 240 (988), $\Delta_f = 24$ (56), and algebraic degree 7 (9). Then, we have linearly transformed these functions to 1-resilient or PC(1) functions. 1-resilient Boolean functions having $\Delta_f < 2^{\frac{n+1}{2}}$ have not been demonstrated earlier for any variable. Further we have obtained an 11-variable RSBF having nonlinearity 990, $\Delta_f = 56$ and algebraic degree 10, which can be linearly transformed to a PC(1) function. For even number of input variables, we have found 10-variable functions having nonlinearity 488, $\Delta_f = 24$ and algebraic degree 9; some of them can be linearly transformed to PC(1) functions.

In [60], a tight upper bound on nonlinearity has been proposed for resilient Boolean functions and the existence of some Boolean functions on 7 to 10 variables satisfying the bound has been posed as an open problem. Since then, the construction of these functions has been a challenging question. Some of them are presented in [51, 37, 68, 58]. The 10-variable 1-resilient function having nonlinearity 492, which we present here, remained unknown till our work [25]; previously, the best achieved nonlinearity was 488 [37].

Considering a Boolean function f as a mapping from $GF(2^n) \to GF(2)$, the functions for which $f(\alpha^2) = f(\alpha)$ for any $\alpha \in GF(2^n)$, are referred to as idempotents [15, 16]. In [52], 15-variable Patterson-Wiedemann functions having nonlinearity $16276 = 2^{15-1} - 2^{\frac{15-1}{2}} + 20$ are identified in the idempotent class. As pointed out in [15, 16], the idempotents can be seen as RSBFs with proper choice of basis. From this motivation, we study the RSBF class and discover 9-variable functions with nonlinearity 241. In [15, 16], the nonlinearity of RSBFs up to 9 variables is studied providing encouraging results. After that, the class of RSBFs has received a lot of attention in the literature [67, 66, 68, 11, 7, 20, 43, 42], where it has been asserted theoretically and experimentally that the RSBF class contains important functions having good cryptographic properties. Further, in [53], RSBFs are studied as components in the rounds of a hashing algorithm and research in this direction was later continued in [10].

Heuristic strategies, such as genetic algorithms and hill climbing, have been initially investigated in [44, 45, 46] for the design of Boolean functions. However, these attempts seem to be insufficient in designing *near-the-best* Boolean functions. After that, simulated annealing [31], a heuristic optimization thechnique based on annealing process of metals, was used to provide promising results [9, 28], which yields Boolean functions having good cryprographic properties in both Walsh and autocorrelation spectra for small functions ($n \leq 8$). On the other hand, in [8], it was pointed out that some of the Boolean functions found by simulated annealing could be linearly transformed (using simple linear change of basis) to achieve resiliency supplying the best possible trade-offs. Consequently, supplementing optimization with theory yields the best Boolean functions on $n \leq 8$ variables in terms of nonlinearity, algebraic degree and resiliency. Recently, exploiting a heuristic strategy based on “Particle Swarm Optimization” [1], the existence of 9-variable, 3-resilient functions having nonlinearity 240 has been demonstrated in [58], which was open since Crypto 2000 [60].

In this chapter, our steepest-descent-like iterative algorithm that first appeared in [29] is suitably modified so that it can be efficiently applied for a search in the RSBF class and attained Boolean functions which are very good in terms of their Walsh and autocorrelation spectra. The strategy presented in [29] has been applied to the complete space of Boolean functions that resulted in discovery of 8-variable balanced Boolean functions f having nonlinearity 116 and $\Delta_f = 16$. It performs much better when applied to the much smaller (but rich) space of RSBFs. To have a quick feel of how efficient our strategy is, one may refer to Remark 1.

In the following section we present the search strategy. The results are presented in Section 3.3.

3.2 Search Strategy

The search strategy uses a steepest-descent-like iterative algorithm, where each iteration step has the input Boolean function f and the output Boolean function f_{min}. At each iteration step, a cost function is calculated within a pre-defined neighborhood of f and the Boolean function having the smallest cost is chosen as the iteration output f_{min}. We use the sum of squared errors [28, 71] as the cost function, which is defined as:

$$Cost = \sum_{\omega} (W_f^2(\omega) - 2^n)^2. \tag{3.1}$$

Note that since $\sum_{d \neq 0} {r_f}^2(d) = 2^{-n} \sum_w ({W_f}^2(w) - 2^n)^2$ (see Theorem 2.1), the cost function minimizes the squared distance to bent functions both in terms of Walsh and autocorrelation spectra.

In some rare cases, the cost of f_{min} may be larger than or equal to the cost of f. This is the crucial part of the search strategy, which provides the ability to escape from local minima and its distinction from the steepest-descent algorithm.

The 1-neighborhood of f is obtained by flipping a single element of its truth table. For an n variable balanced Boolean function, the 1-neighborhood consists of 2^n many distinct Boolean functions, each being at the Hamming distance 1 to the original Boolean function. However, when the search space is restricted to RSBFs, the 1-neighborhood is either an empty set or contains a single RSBF. If a bit in the truth table of an RSBF is changed, all entries corresponding to an orbit (a rotationally symmetric partition, which is composed of vectors that are equivalent under rotational shifts) should be changed to obtain another RSBF. The closest rotationally symmetric neighbors of RSBFs can be found by complementing the truth table entries corresponding to a complete orbit. So, at each step of the algorithm, we constitute the neighborhood of f by complementing each RSTT entry (i.e., changing all the values in a truth table corresponding to an orbit).

Our steepest-descent-like search technique minimizes the cost until a local minimum is attained, then it takes a step in the direction of non-decreasing cost. That is, whenever possible, the cost is minimized; otherwise, a step in the reverse direction is taken. The deterministic step in the reverse direction corresponds to the smallest possible cost increase within the pre-defined neighborhood of the preceding Boolean function, which makes it possible to escape from the local minima.

Our algorithm given below starts with an arbitrary RSBF, $f_{initial}$, and stops after a fixed number of iterations, N. At each iteration, g_n distinct Boolean functions within the predefined neighborhood, each of which is shown by $f_{flipped}$, are visited by storing the cost value $cost_{flipped}$ in $COST$, and the corresponding Boolean function itself in SET_f. Among the stored cost values, the minimum one, $cost_{min}$, is chosen, and the respective Boolean function, f_{min}, is obtained from SET_f as the candidate of the step output. If the candidate f_{min} is already in $STORE$, which contains all previous iteration outputs, then this candidate f_{min} and its cost are removed from SET_f and $COST$ respectively. The minimum cost value is searched again in $COST$ among the remaining cost values to find the respective new candidate for f_{min}.

```
Algorithm 3.1
$f = f_{initial}$;
for(int $k = 0$; $k < N$; $k++$){
    for(int $i = 0$; $i < g_n$; $i++$){
        Flip one orbit of $f$
        $SET_f[\ i\ ] = f_{flipped}$
        $COST[\ i\ ] = cost_{flipped}$
    }
    Find $cost_{min}$ (min. $cost_{flipped}$ in $COST$), and $f_{min}$ (respective $f_{flipped}$ in $SET_f$)
    while($f_{min}$ is already in $STORE$){
        Remove $cost_{min}$ from $COST$, and $f_{min}$ from $SET_f$
        Find $cost_{min}$ in $COST$, and $f_{min}$ in $SET_f$
    }
    $STORE[k] = f_{min}$
    $f = f_{min}$
}
```

Since the neighbors of f are obtained simply by flipping a bit in its RSTT, the number of neighbors is equal to g_n. We present the C code of Algorithm 3.1 in Appendix C.

3.3 Results

We start this section with the most important result of this chapter.

3.3.1 9-variable RSBF with nonlinearity 241

The following is the truth table of a 9-variable function $f(x_1, \ldots, x_9)$ having nonlinearity $2^{9-1} - 2^{\frac{9-1}{2}} + 1 = 241$.

```
977F3FFA0EFAAEC955F8FACDCCA9A0837666EBC0FA88E0B3F4E08983C845915E
7F7C2C29FCCBA101EA98C085E8118B5EFE21E9118483851EE1952136971676E9
```

Given an integer $m > 0$ and even, it is clear that the function $g(y_1, y_2, \ldots, y_m) \oplus f(x_1, \ldots, x_9)$ is an n-variable ($n = m+9$) function with nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 2^{\frac{m}{2}})$, where $g(y_1, y_2, \ldots, y_m)$ is an m-variable bent function. Thus there exist Boolean functions having nonlinearity $>$ $(2^{n-1} - 2^{\frac{n-1}{2}})$ for odd $n \geq 9$. Keeping this in mind, and adding the results of [2, 47] that the maximum nonlinearity of Boolean functions on odd number of variables for odd $n \leq 7$ is $(2^{n-1} - 2^{\frac{n-1}{2}})$, we get the following.

Theorem 3.1. There exist Boolean functions on n (odd) variables having nonlinearity $>$ $(2^{n-1} - 2^{\frac{n-1}{2}})$ if and only if $n > 7$. In other words, for odd n, the covering radius of the $(2^n, n+1)$ Reed-Muller code is $> (2^{n-1} - 2^{\frac{n-1}{2}})$ if and only if $n > 7$.

Remark 3.1. At this point we like to highlight the efficiency of the search method. In [24] (our work later to [25]), it has been noted that there are 1512 many 9-variable RSBFs having nonlinearity 241 and this is the maximum nonlinearity in the 9-variable RSBF class. Note that the 9-variable RSBF class is of size 2^{60}. Thus in a random search, the probability of getting a 9-variable RSBF with nonlinearity 241 is $1 - (1 - \frac{1512}{2^{60}})^i$ in i many attempts. Note that $\lim_{z\to\infty}(1 - \frac{1}{z})^z = \frac{1}{e}$. Thus in approximately $\frac{2^{60}}{1512}$ ($> 2^{49}$) many attempts one may get a 9-variable RSBF having nonlinearity 241 with probability $1 - \frac{1}{e} > \frac{1}{2}$ in a random search. Our search method performs much better than that. We found 5 many RSBFs having nonlinearity 241 in $2 \cdot 10^8$ ($< 2^{28}$) many generation of Boolean functions using Algorithm 3.1; which shows that the efficiency of our search strategy plays an important role to discover such functions.

Next we concentrate on other important functions in the RSBF class.

3.3.2 Important RSBFs on 9, 10 and 11 variables

In Table 3.1, we summarize the profiles of the some other important RSBFs that we obtain by Algorithm 3.1. We use the notation (number of variables, resiliency, degree, nonlinearity, absolute indicator) for each profile; resiliency = -1 (respectively 0) denotes unbalanced (respectively balanced) functions. If the given profile $(n, m, d, \sigma, \Delta)$ can be transformed into a function having the property of PC(1), then we denote it by $(n, m, d, \sigma, \Delta)$*.

Table 3.1: Summary of other important RSBFs.

RSBF No	Initially Attained RSBF	Affinely Transformed Results
1	(9, 0, 7, 240, 24)	(9, 0, 7, 240, 24)*, (9, 1, 7, 240, 24)
2	(11, -1, 10, 990, 56)*	(11, 0, 10, 990, 56)*
3	(11, 0, 9, 988, 56)	(11, 0, 9, 988, 56)*, (11, 1, 9, 988, 56)
4	(10, -1, 9, 488, 24)*	(10, 0, 9, 488, 24)*
5	(10, -1, 8, 492, 56)	(10, 1, 8, 492, 56)

1. Algorithm 3.1 outputs the following function ϕ, which is a 9-variable balanced RSBF having $nl(\phi) = 240$ and $\Delta_\phi = 24 < 32 = 2^{\frac{9+1}{2}}$ and algebraic degree 7.

```
005473257A0E49676BDD10E864D3287F399BB2E30214BC916865E70B58853BBE
0ED3C29B9F48AD0F554906658BB1C3562D857833F92B159E33C5D1765BDEDEE9
```

Given an n-variable Boolean function f, let us form a set of vectors S_f, which is defined as

$$S_f = \{\omega \in \{0,1\}^n \mid W_f(\omega) = 0\}. \tag{3.2}$$

Suppose B_f is a nonsingular $n \times n$ matrix whose rows are linearly independent vectors belonging to S_f (if there exist such vectors in S_f). Then, defining $f'(x) = f(B_f^{-1}x)$, it is ensured that both f' and f *have the same weight, nonlinearity and algebraic degree* [33]. Moreover, $W_{f'}(\omega) = 0$ for $wt(\omega) = 1$, which provides that f' is correlation immune of order 1. Further if f is balanced then f' is 1-resilient, in other words, if $W_f(0) = 0$, then $W_{f'}(0) = 0$ for $0 \le wt(\omega) \le 1$. This technique has been used in [50, 38, 8]. The following function is obtained by a linear transformation on the input variables of ϕ above, which is 1-resilient.

```
1C969EEC0B5B87307EB530AD3C365AD32A6771C130CBA71435798C8B6A9DE615
ECF9D05D64E8987F8414D1018621E7EEE05FD4E1AF403F05BF2226AEE2B36D0E
```

Similar technique can be used to construct PC(1) functions. Given an n-variable Boolean function f, let us define $T_f = \{\alpha \mid \Delta_f(\alpha) = 0\}$. If there exist n linearly independent vectors in T_f, then one can form a nonsingular $n \times n$ matrix D_f whose rows are linearly independent vectors belonging to T_f. Now one can define $f'(x) = f(xD_f)$. Both f' and f *have the same weight, nonlinearity and algebraic degree* [33]. Moreover, $\Delta_{f'}(\alpha) = 0$ for $wt(\alpha) = 1$. This ensures that f' is PC(1). This technique has been used in [39]. The following function is obtained by a linear transformation on the input variables of ϕ above, which is PC(1).

```
2C317F8130464E9D30EA0A95556F8EAAE108188979AC48E9F23AA6793CBBE526
F0DA686073CFD3D6ABE78F641FEB34DD64ED3721BCE0C6CA0CB8E5FCA6655004
```

It would be interesting to get a transformation on input variables such that 1-resiliency and PC(1) can both be achieved at the same time.

2. Using Algorithm 3.1 we find the following function ϕ, which is an 11-variable unbalanced RSBF having $nl(\phi) = 990$, $\Delta_\phi = 56$, and $deg(\phi) = 10$. Note that this function is by itself a PC(1) function which is not balanced, but soon we will provide a balanced PC(1) function too.

```
FEEDB8A7CA94D83AF4C88330F7C04EC8BB64F4C5C05B0F41BB6AF41130BCB595
CACF7D60FF75F463B04473DB00FE2553DACF7CDDAE6517161A40DAA08A32D263
F198E0EE3FA62C15BEFE3A36BF75280A8B5571703A1EE7CA4551BEEC4C23725A
A798A4BF2EB5B3A6C9FC7C63566A562806996510A2D8984484CC1B49B60D684B
EB4386C4E814F8A85AEB8D3958E546778BF8FFE94ADD0E3DCBEF2B7648C004C9
D48A72276E467F001FDC46B8BD6AA1CC342727529EE9E8E025B40C4A2A596389
992A86C0C935CBAF1CF98F279B1E8829E0C3AAF07EA4781A633C698836280D91
502897936D335601890CE2C496906035C075B5E1128A64878F7940A33D8171DE
```

We transform ϕ to a balanced one, f, by using $\omega = (0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1)$ for which $W_\phi(\omega) = 0$. Thus the function $f(x) = \phi(x) \oplus \omega \cdot x$ is balanced as given below with the same $nl(\phi)$, Δ_ϕ, and $deg(\phi)$.

```
9784D1CE5C024EAC625E15A69EA927A1D20D9DAC56CD99D72DFC628759D5DCFC
A3A6140969E362F526D2E54D69974C3AB3A615B438F381808CD64C36E35BBB0A
98F18987A930BA832868ACA0D61C4163E23C1819AC88715CD3C7287A254A1B33
```

```
CEF1CDD6B82325305F6AEAF53F033F416FF00C79344E0ED2125A8DDFDF640122
822AEFAD7E826E3ECC7D1BAF318C2F1EE2919680DC4B98AB5D79BDE021A96DA0
BDE31B4EF8D0E996894AD02ED403C8A55D4E4E3B087F7E76B3229ADC43300AE0
F043EFA95FA35D398A6F19B1F277E14089AAC399E832EE8CF5AAFF1E5F4164F8
3941FEFAFBA5C0971F9A7452FFF9095CA91CDC88841CF21119EFD63554E818B7
```

Note that this balanced function f is by itself a PC(1) function. Since the above function is of nonlinearity 990, which is not divisible by 4, it cannot be made 1-resilient by affine transformation.

3. Next we present such a function of nonlinearity 988 having degree 9, so that we may get 1-resiliency by linear transformation on input variables. By Algorithm 3.1, we get a balanced RSBF f having nonlinearity 988, $\Delta_f = 56$ and algebraic degree 9.

```
ECB4DE71F3FD6B13FB1ABAB7688A075EFA9F17D89B9DCA3E6D80D0CC542B63ED
BE8992BB076FE6C083CAD2A7E0CD4AE96CE6C411A244F4B166600D9F281AB8B6
DEB881879619CBCB407E29BAFC3CE501C14AB0DCA31CCD2BEC01F4A621C8E8D7
7DF1FD28B0201317CC5C3421EB618F533969280455B2D3BB4DD04299CF859F7C
A6EDDF95D447803EC77C5786B0CAE19B61453BA818C38B89AAA50AE5A8370446
E41365998E14E6B1984E16B1A1E2188FFDF04057AE61993D5902F4D5BC85B37E
2BF7EB06BEF248959B504911030B072AA5B526A54A651843FC8F2957D0EF635E
1F926C875C95113037238B49E31FCB9E74A3E75471199796E1BED57696FE6EA0
```

The function is then transformed to 1-resilient function as follows.

```
975D2EFDA7C9D97E96B58F09B056960188614907BACF4617219BF147E6B34314
410C9E8BB000FE87E8A7A3590CF4B1A66D11818429EC3F0F61EF89CB9E898BE0
B208B29527E8404F871B756693944C3972D242039F3017FD34E2973C2B2567A5
C2FFF57B3783DD747993E8346E5DE671ADE80D4D3E98FA461ACAA93A2FF87622
D0BCA271ABDD139C66ED2D8C75ED7DD3B22968E85BC520361B31DD9F09FF1162
974F19DD09251C16C56CDC7C3AE920EADBC08BFC51B3F300DE3C7B6CC668D504
01EC68AC1D3AB7525BEEE63D0C208D358F88DED59DD59B4433B80016AF5DA8BD
D8E2B053C0E67A16241122E8E4A4C158CB654ABAFDA03E73A05A75DA610B99BF
```

Further we can also transform f to a PC(1) function as follows.

```
850EC14AF195F38DD59EB29E7CD758C76122F20FCE9E83DE393F53757954269F
44C0EC07E6724883E726A750939EE4DB5475F56C1D3933F585C6DB9719D8BA35
58041ABEF105914D59F02FFB8CED823D982469B85F32874654BB8CBAEB4A110E
F2381C97099C58E1A0FD724A35D28129D9F61CA877BE0109BD67A3B62EA4BA5F
0DA4AE0E2D84AA64301635E183CE33D19B7D50C9230D027BBB22443BF5765A34
F3AF2B9F0AB2DE85ABDF1367526B942351D91F43EB123B9CE5B164E6DFB95E81
60D537FB9ED65A0AD8674BC3443C83804D5D3169CD0E5B22E723184D144D0918
00332B98CF8E2E39F53C6BAEF24402F19B9B703616B1C860AE538705DEEAF0B7
```

4. Using Algorithm 3.1, We first find an unbalanced RSBF ϕ with $nl(\phi) = 488$, $\Delta_\phi = 24$, and $deg(\phi) = 9$ as given below. This is an unbalanced PC(1) function.

```
FFFEEBF9E8CAAFD2E8C5A4899CFFB20CFDC4F162992580C283E5FAAA8F1C51B5
FAA6B471FA12385996824D379154A55DD10EA827BF9D8D98D0EB07B43606CE27
FE9D883C8B216F42FAD853081BC036D7C26DC44D60B75E3FD2037734C93662A3
E70611B8CCD0586F8BAB87E7C1F69681B254ACCB113B9E614E295569A1F91D7F
```

We find only one zero at $\omega = (1,1,1,1,1,1,1,1,1,1)$ in the Walsh spectrum of ϕ, which is used to get a balanced function as follows. One may note that this function is by itself a PC(1) function.

```
96687D907EA3C6447EACCD1FF56924656BAD98F4F0B316ABEA736CC319753823
6CCFDDE79384AE30FF14DB5E073DCCCBB8983E4E29F4E40E46826E225F90584E
68F4E1AAE2B7F92B934EC5618DA95F41ABFB5224F6DE37A9446A1EA2A0A0F4CA
8E9087D15AB931F91DC2EE71A86000E8243DC55D78AD080827BFC300379074E9
```

5. We first identify unbalanced RSBF ϕ having nonlinearity 492 and algebraic degree 8. The function ϕ is as follows.

```
E9C6B17C9F136FE496BA574B7CEEA820D33C8E9D776F709B6EB1A8E9CCD01941
B34F4EF095F8C2E23E6A68AA6B40C2DA3CE8DB469C81A883F4A1A24146877153
9A5E75BA64F9EA00D627FBC5A509AC595BAC7C886880988C68DA6101E109A3DD
4EF4AD80E3DB312DD2E080428C91911FAE309D53C8082557247D803F2F07335E
```

To make it balanced we take $\omega = (0,0,0,0,0,0,0,1,0,1)$ where $W_\phi(\omega) = 0$. Thus $f = \phi \oplus \omega \cdot x$ is a balanced function. Then we consider the set $S_f = \{\omega \in \{0,1\}^n \mid W_f(\omega) = 0\}$ having $|S_f| = 40$. There exist 10 linearly independent vectors in S_f, and one can construct a nonsingular 10×10 matrix B_f whose rows are linearly independent vectors from S_f. We have considered the following matrix.

$$B_f = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.3}$$

Let, $C_f = B_f^{-1}$ and then $f'(x) = f(C_f x)$ is a 10-variable 1-resilient function with algebraic degree 8 and nonlinearity 492. The function f' is as follows.

```
8180CDED6C1C0302AA32E761B2079F0C37D8393E5B8DF2934B2AACEA7EB40BF0
AF6694BAF19E415E4580C0D679DB9BEB982963591185C33FEC2F67987D121D3B
C4E281F3D071957A74DF8A99FF258E9EC3D3AE6BE39415B0F4E5DA104DFC0125
24AD19CBA965D3768C525AD75C5316AA0F77F1A49E4AFD4223D40756C8388886
```

3.3.3 Search Effort

In Remark 1, we have quantified how efficient our search method is in terms of finding 9-variable RSBFs having nonlinearity 241. Since it is not possible to completely enumerate the other important functions we have achieved, the efficiency of the search method cannot be quantified. However, the following search effort related to Algorithm 3.1 show that it is indeed possible to achieve good functions in nominal time.

For $n = 9$ we have carried out 2000 runs each with $N = 100,000$ iterations. Among these 200 million RSBFs, five have the nonlinearity 241, and 580 many RSBFs have the nonlinearity 240 and absolute indicator 24. For $n = 10$, 250 runs have been performed each with $N = 400,000$ iterations. Among the total of 100 million RSBFs, 11 have the nonlinearity 488 and absolute indicator 24, all transformable to balanced functions. In the same experiment, we have found 67,479 RSBFs with nonlinearity 492, all transformable to balanced functions and among them we could obtain several 1-resilient functions using linear change of basis. Besides, we have noticed that only four of the 67,479 RSBFs are balanced, and none of these balanced functions can be transformed into a 1-resilient function. For $n = 11$, there are 7 successes with nonlinearity 988 and absolute indicator 56 in 500 runs. Moreover, we have encountered an unbalanced RSBF having nonlinearity 990 and absolute indicator 56, which is transformable to a balanced function.

Using a computer system with Pentium IV 2.8 GHz processor and 256 MB RAM having Windows XP operating system, and setting the iteration number $N = 100,000$, a typical run of our algorithm takes 1 minute and 29 seconds for $n = 9$. With the same computer system, a typical run takes 57 minutes for $n = 10$, and 69 minutes for $n = 11$, by setting the iteration numbers to $N = 400,000$ and $N = 500,000$ respectively.

3.4 Conclusions

Boolean functions, which have not been known for a long time, could be achieved with our steepest-descent-like iterative heuristic search in the class of rotation symmetric Boolean functions. As a major result, we find a 9-variable RSBF with nonlinearity 241 and thus we could show the existence of Boolean functions having nonlinearity $> (2^{n-1} - 2^{\frac{n-1}{2}})$ for $n = 9, 11, 13$. We could find balanced Boolean functions on $9, 10$ and 11 variables with maximum absolute value in the autocorrelation spectrum $< 2^{\lceil \frac{n}{2} \rceil}$ with other cryptographic properties such as good nonlinearity and algebraic degree. Some of these functions on each of the $9, 10$ and 11 variables cases can be affinely transformed to balanced PC(1) functions. Some of these functions on 9 and 11 variables can be transformed to 1-resilient functions as well. Further, we discovered several 10-variable 1-resilient functions with nonlinearity 492, which was posed as an open question in Crypto 2000.

CHAPTER 4

9-VARIABLE RSBFs WITH NONLINEARITY $>$ 240

4.1 Introduction

In this chapter, the complete class of 9-variable RSBFs is studied for nonlinearity 241. For this purpose, we exploit the nice combinatorial structure of the Walsh spectra for RSBFs on odd number of variables [43] to carry out the exhaustive search of 9-variable RSBFs having nonlinearity $>$ 240 with considerable computational effort. As a consequence, we find that there are 8×189 many RSBFs having nonlinearity 241 that is maximum possible in the class. Further, utilizing some variants of binary nonsingular circulant matrices, it is shown that there are only two different 9-variable RSBFs having nonlinearity 241 up to the affine equivalence.

In the following section, the details of the exhaustive search strategy for finding 9-variable RSBFs having nonlinearity $>$ 240 is explained. The affine equivalence among 9-variable RSBFs having nonlinearity 241 is presented in Section 4.3.

4.2 Exhaustive Search Strategy

First we permute the orbit leaders to obtain the matrix $\mathcal{A}^{\pi}$ as discussed in Section 2.1. Given the new ordering of Λ_i's, let us represent two vectors

$$\mu_f = ((-1)^{f(\Lambda_0)}, \ldots, (-1)^{f(\Lambda_{\frac{g_n}{2}-1})}) \text{ and } \nu_f = ((-1)^{f(\Lambda_{\frac{g_n}{2}})}, \ldots, (-1)^{f(\Lambda_{g_n-1})}) \qquad (4.1)$$

corresponding to an n-variable RSBF f, where $g_n = |\mathcal{G}|$. Then, considering the vectors $u_f = \mu_f\, \mathcal{H}$, $v_f = \nu_f\, \mathcal{H}$ and denoting their i-th ($i = 0, \ldots, \frac{g_n}{2} - 1$) components by $u_f[i]$, $v_f[i]$ respectively, it is seen that the vaules in the Walsh spectrum of f can be calculated as $(u_f[i] + v_f[i])$ for the first $\frac{g_n}{2}$ many representative elements (which are of even weights) and $(u_f[i] - v_f[i])$ for the next $\frac{g_n}{2}$ many representative elements (which are of odd weights). For 9-variable RSBFs, the matrix $\mathcal{A}^{\pi}$ is a 60×60 matrix, as the number of distinct orbits $g_n = 60$; hence, the matrix $\mathcal{H}$ is a 30×30 matrix. We start with a technical result which is easy to prove.

Proposition 4.1. Let a, b and M be three integers with $M > 0$. Then $|a+b| \leq M$, $|a-b| \leq M$ iff $|a| + |b| \leq M$.

To achieve nonlinearity > 240, it is clear that the Walsh spectrum values of a Boolean function must be in the range $[-30, 30]$. Keeping this in mind and adding the result of Proposition 4.1, one can see that $|u_f[i]| + |v_f[i]| \leq 30$, where $0 \leq i \leq \frac{g_9}{2} - 1 = 29$, for a 9-variable RSBF (represented by a 60-bit vector $\mu_f || \nu_f$) having nonlinearity > 240. A naive method to extract such RSBFs requires to generate all the $\mu_f || \nu_f$ patterns, which exhausts the search space of 2^{60}.

We present an efficient method for this purpose. As each of the patterns μ_f and ν_f must satisfy the necessary conditions $|u_f[i]| \leq 30$ and $|v_f[i]| \leq 30$ respectively for $0 \leq i \leq 29$, we first search for all the patterns μ_f's such that $|u_f[i]| \leq 30$ for $0 \leq i \leq 29$. Note that the search for all the patterns ν_f's such that $|v_f[i]| \leq 30$ for $0 \leq i \leq 29$ produces the same result. Let us represent the set of resulting patterns by S. By fixing $\mu_f[0] = (-1)^0 = 1$ (or $\nu_f[0] = (-1)^0 = 1$), this search requires the generation of 2^{29} many patterns of μ_f (or ν_f). The reason why we fix the first bit of the patterns will be explained by Proposition 4.2. Using a computer with the specification 3.6 Ghz Intel Xeon and 4 GB RAM, it takes little less than half an hour to obtain the set S and it contains 24037027 ($< 2^{25}$) many patterns. Consequently, the search for 9-variable RSBFs having nonlinearity greater than 240 reduces to the problem of choosing any two patterns μ_f, ν_f from S such that the resulting concatenation $\mu_f || \nu_f$ satisfies the condition $|u_f[i]| + |v_f[i]| \leq 30$ for $0 \leq i \leq 29$. Let us first present the following technical result, which helps us explain how we select two patterns from the set S.

Proposition 4.2. Consider a 9-variable RSBF f which is represented as $\mu_f || \nu_f$ such that $|u_f[i]| + |v_f[i]| \leq 30$ for $0 \leq i \leq 29$. Let $l_8 = x_0 \oplus x_1 \ldots \oplus x_7 \oplus x_8$, the rotation symmetric linear function containing all the variables. Consider the functions g such that any of the following holds:

$$\mu_g = \mu_f, \nu_g = \nu_f^c, \text{ i.e., } g(x_0 \ldots x_8) = f(x_0, \ldots, x_8) \oplus l_8, \tag{4.2a}$$

$$\mu_g = \mu_f^c, \nu_g = \nu_f, \text{ i.e., } g(x_0 \ldots x_8) = f(x_0, \ldots, x_8) \oplus l_8 \oplus 1, \tag{4.2b}$$

$$\mu_g = \mu_f^c, \nu_g = \nu_f^c, \text{ i.e., } g(x_0 \ldots x_8) = f(x_0, \ldots, x_8) \oplus 1, \tag{4.2c}$$

$$\mu_g = \nu_f, \nu_g = \mu_f, \text{ i.e., } g(x_0 \ldots x_8) = f(1 \oplus x_0, \ldots, 1 \oplus x_8), \tag{4.2d}$$

$$\mu_g = \nu_f, \nu_g = \mu_f^c, \text{ i.e., } g(x_0 \ldots x_8) = f(1 \oplus x_0, \ldots, 1 \oplus x_8) \oplus l_8, \tag{4.2e}$$

$$\mu_g = \nu_f^c, \nu_g = \mu_f, \text{ i.e., } g(x_0 \ldots x_8) = f(1 \oplus x_0, \ldots, 1 \oplus x_8) \oplus l_8 \oplus 1, \tag{4.2f}$$

$$\mu_g = \nu_f^c, \nu_g = \mu_f^c, \text{ i.e., } g(x_0 \ldots x_8) = f(1 \oplus x_0, \ldots, 1 \oplus x_8) \oplus 1, \tag{4.2g}$$

Then $|u_g[i]| + |v_g[i]| \leq 30$ for $0 \leq i \leq 29$.

Hence, for a single 9-variable RSBF f there exist 8 many (including f) affinely equivalent RSBFs providing the same nonlinearity. Hence, the reason we fix $\mu_f[0] = 1$ is to remove such affinely equivalent RSBFs from S. Initially we note that, by Parseval's theorem, repeating a pattern from S twice (i.e., $\mu_f||\nu_f$, when $\nu_f = \mu_f$) one can not satisfy the condition $|u_f[i]| + |v_f[i]| \leq 30$ for $0 \leq i \leq 29$; in such a case, the maximum possible nonlinearity is 240. Thus $\binom{24037027}{2} = 288889321480851$ ($< 2^{49}$) many pairs are needed to check.

To reduce the number of patterns in S, a sieving method is then applied as follows. For some t, $0 \leq t \leq 29$, all the μ_f patterns in S satisfying $|u_f[t]| = 30$ are stored in the set $S_{30,t}$. Similarly, the set $S_{0,t}$ is constituted for the ν_f patterns satisfying $|v_f[t]| = 0$. After that, selecting each of the μ_f patterns from $S_{30,t}$ and each of the ν_f patterns from $S_{0,t}$, we check the condition $|u_f[i]| + |v_f[i]| \leq 30$ for all i, $0 \leq i \leq 29$. If it holds, we store $\mu_f||\nu_f$, which yields a 9-variable RSBF having nonlinearity 241. Note that since the μ_f patterns in $S_{30,t}$ cannot be concatenated with any ν_f patterns in S except the ones in $S_{0,t}$, to achieve nonlinearity 241, the set S is updated by $S \setminus S_{30,t}$ for each t.

In the process of applying the sieving method, the following observations are encountered.

1. For $t = 0$ the set $S_{30,t}$ is an empty set; so we do not consider this.

2. For $t = 28$ there is no ν_f pattern such that $|v_f[28]| \leq 2$; hence, we initially update the set S removing all the μ_f patterns such that $28 \leq |u_f[28]| \leq 30$, which reduces $|S|$ from 24037027 to 18999780.

Table 4.1 shows the number of patterns $\mu_f||\nu_f$ yielding 9-variable RSBFs having nonlinearity 241 for each t (except $t = 0$ and 28). It is seen that the sieving method provides $7 \times 27 = 189$ many 9-variable RSBFs having nonlinearity 241, and hence, we get 8×189 many 9-variable RSBFs having the same nonlinearity. This experiment reduces S from 24037027 to 9540580. Using the same computer system, the experiment requires little more than a day.

Table 4.1: Initial search result for 9-variable RSBFs having nonlinearity 241.

t	$\|S_{30,t}\|$	$\|S_{0,t}\|$	# of $\mu_f\|\|\nu_f$ such that $nl(f) = 241$
1	687215	37584	0
2	514474	37584	0
3	132406	77328	27
4	545152	37584	0
5	408014	37584	0
6	255915	37584	0
7	126821	77328	27
8	338321	37584	0
9	206952	37584	0
10	237525	37584	0
11	121290	77328	27
12	464475	37584	0
13	364029	37584	0
14	385125	37584	0
15	552651	77328	27
16	531456	37584	0
17	222237	37584	0
18	115705	77328	27
19	495350	37584	0
20	272767	37584	0
21	192113	37584	0
22	104643	77328	27
23	320685	37584	0
24	597941	37584	0
25	110174	77328	27
26	542078	37584	0
27	613686	37584	0
29	747073	37584	0

Then we check $\binom{9540580}{2}$ ($< 2^{46}$) many pairs, which takes 30 hours using 20 computers in parallel, each with the specification of 2.8 GHz Pentium IV and 256 MB RAM having Windows XP operating system. Finally, we do not find any other RSBF having nonlinearity > 240, suggesting the following result.

Theorem 4.1. There are 8×189 many 9-variable RSBFs having nonlinearity 241 and this is the highest nonlinearity for the 9-variable RSBF class.

Interestingly, all of the 189 many 9-variable RSBFs having nonlinearity 241 available from Table 4.1 have the same distribution of Walsh spectra, which is presented in the following table.

Table 4.2: Distribution of Walsh spectra of the functions found in Table 4.1.

$W_f(\omega)$	-30	-22	-14	-6	2	10	18	26
# of ω's	127	27	36	18	55	39	54	156

Among these RSBFs, we find that there are only two classes (63 of them in one class and the rest in another class) having different distribution of autocorrelation spectra as can be seen from the following table.

Table 4.3: Distribution of autocorrelation spectra of the functions found in Table 4.1.

$r_f(d)$	-52	-44	-36	-20	-12	-4	4	12	28
# of nonzero d's	9	9	9	18	81	85	198	81	21

$r_f(d)$	-76	-36	-28	-20	-12	-4	4	12	20	28
# of nonzero ω's	1	9	18	36	81	135	108	54	48	21

Thus it seems that among the 189 RSBFs found in Table 4.1 there are only two different RSBFs up to the affine equivalence, which is justified in the next section.

4.3 Affine equivalence of RSBFs having nonlinearity 241

Let f and g be Boolean functions on n variables. We call them affinely equivalent if the following condition is satisfied

$$g(x) = f(xA \oplus b) \oplus d \cdot x \oplus c, \tag{4.3}$$

where A is an $n \times n$ binary nonsingular matrix, b, d are n-bit binary vectors and c is a binary constant. Note that, in Proposition 4.2 the g functions are affinely equivalent to f.

Given $(a_0, \ldots, a_{n-1}) \in \{0,1\}^n$, the $n \times n$ circulant matrix generated by $(a_0, \ldots, a_{n-1})$ is in the form

$$C(a_0, a_1, \ldots, a_{n-1}) = \begin{bmatrix} a_0 & a_1 & a_2 & \ldots & a_{n-1} \\ a_{n-1} & a_0 & a_1 & \ldots & a_{n-2} \\ a_{n-2} & a_{n-1} & a_0 & \ldots & a_{n-3} \\ \vdots & & & & \vdots \\ a_1 & a_2 & a_3 & \ldots & a_0 \end{bmatrix}. \tag{4.4}$$

The determinant of the matrix $C(a_0, a_1, \ldots, a_{n-1})$ is given by

$$det[C(a_0, a_1, \ldots, a_{n-1})] = \prod_{i=0}^{n-1} (a_0 + a_1\zeta_i + a_2\zeta_i^2 + \ldots + a_{n-1}\zeta_i^{n-1}), \tag{4.5}$$

where ζ_i's $(0 \leq i \leq n-1)$ are the n-th roots of unity. In particular we denote $\zeta_0 = 1$. We are interested in the binary circulant matrices which are nonsingular.

Proposition 4.3. Let $\alpha, \beta \in \{0,1\}^n$ such that $\alpha \in G_n(\beta)$ and A be an $n \times n$ nonsingular binary circulant matrix. Then $\alpha A \in G_n(\beta A)$.

Proof. As $\alpha \in G(\beta)$, we have $\alpha = \rho^k(\beta)$, for some k such that $0 \leq k < n$. It is also clear that the columns $A_1, A_2, \ldots, A_n$ of the matrix $A = C(a_1, a_2, \ldots, a_n)$ are cyclic shift of each other, precisely, $A_j = \rho^{j-1}(A_1)$. Now,

$$\begin{aligned} \beta A &= (\beta A_1, \beta A_2, \beta A_3, \ldots, \beta A_n), \\ &= (\beta A_1, \beta\rho^1(A_1), \beta\rho^2(A_1), \ldots, \beta\rho^{n-1}(A_1)), \\ &= (\beta A_1, \rho^{n-1}(\beta)A_1, \rho^{n-2}(\beta)A_1, \ldots, \rho^1(\beta)A_1). \end{aligned} \tag{4.6}$$

Again,

$$\begin{aligned} \alpha A &= (\alpha A_1, \alpha A_2, \alpha A_3, \ldots, \alpha A_{k+1}, \alpha A_{k+2}, \ldots, \alpha A_n), \\ &= (\alpha A_1, \rho^{n-1}(\alpha)A_1, \rho^{n-2}(\alpha)A_1, \ldots, \rho^{n-k}(\alpha)A_1, \rho^{n-k-1}(\alpha)A_1, \ldots, \rho^1(\alpha)A_1), \\ &= (\rho^k(\beta)A_1, \rho^{n-1}(\rho^k(\beta))A_1, \rho^{n-2}(\rho^k(\beta))A_1, \ldots, \rho^{n-k}(\rho^k(\beta))A_1, \\ &\quad \rho^{n-k-1}(\rho^k(\beta))A_1, \ldots, \rho^1(\rho^k(\beta))A_1), \end{aligned}$$

$$
\begin{aligned}
&= (\rho^k(\beta)A_1, \rho^{n-1+k}(\beta)A_1, \rho^{n-2+k}(\beta)A_1, \ldots, \rho^{n-k+k}(\beta)A_1,\\
&\quad \rho^{n-k-1+k}(\beta)A_1, \ldots, \rho^{1+k}(\beta)A_1),\\
&= (\rho^k(\beta)A_1, \rho^{k-1}(\beta)A_1, \rho^{k-2}(\beta)A_1, \ldots, \beta A_1, \rho^{n-1}(\beta)A_1, \ldots, \rho^{k+1}(\beta)A_1). \qquad (4.7)
\end{aligned}
$$

This shows $\alpha A \in G_n(\beta A)$.

Proposition 4.4. Let $f(x)$ be an n-variable RSBF and A be an $n \times n$ nonsingular binary circulant matrix. Then $f(xA)$ is also an RSBF.

Proof. Let $g(x) = f(xA)$. Consider $x_1, x_2 \in G_n(\Lambda)$. Now $g(x_1) = f(x_1A)$ and $g(x_2) = f(x_2A)$. As $x_1A, x_2A \in G_n(\Lambda A)$ (from Proposition 4.3) and f is an RSBF, $g(x_1) = f(x_1A) = f(x_2A) = g(x_2)$. Thus g is also an RSBF.

It is found that there exist 21 different nonsingular binary circulant 9×9 matrices up to equivalence corresponding to the row permutations. Then, using any of these 21 matrices, the 189 RSBFs (available from Table 4.1) are classified (based on Proposition 4.4) into 9 classes each consisting of 21 affinely equivalent RSBFs. More specifically, in a class, the RSBFs are generated as $f(x)$, $f(xA)$, $f(xA^2), \ldots, f(xA^{20})$, where A is one of the 21 matrices, and $f(x)$ is one of the 189 RSBFs. One example circulant matrix generated by $(0, 0, 0, 1, 0, 1, 1, 1, 1)$ is given below:

$$
A = C(0,0,0,1,0,1,1,1,1) = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1\\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1\\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1\\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1\\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0\\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1\\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0\\
0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0\\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0
\end{bmatrix}. \qquad (4.8)
$$

As a consequence, there are 9 representative RSBFs. We have checked that out of these 9 RSBFs, three RSBFs follow the distribution of autocorrelation spectrum presented in the top sub-table of Table 4.3 and six RSBFs follow the distribution of the autocorrelation spectrum presented in the bottom one of Table 4.3.

To achieve further affine equivalence, we consider some larger class of nonsingular matrices than the binary circulant matrices. In particular, instead of starting with a row and then rotate the row one place (we use the right rotation) to generate the next row, we consider that given the first row, we may go for i-rotation such that i, n are coprime.

Let us define $C^i(a_0, a_1, \ldots, a_{n-1})$ as the matrix formed by taking $(a_0, a_1, \ldots, a_{n-1})$ as the first row and each of the other rows is the i-rotation of its preceding row, i.e.,

$$C^i(a_0, a_1, \ldots, a_{n-1}) = \begin{bmatrix} a_0 & a_1 & a_2 & \ldots & a_{n-1} \\ a_{n-i} & a_{n+1-i} & a_{n+2-i} & \ldots & a_{n+n-1-i} \\ a_{2n-2i} & a_{2n+1-2i} & a_{2n+2-2i} & \ldots & a_{2n+n-1-2i} \\ \vdots & & & & \vdots \\ a_{(n-1)n-(n-1)i} & a_{(n-1)n+1-(n-1)i} & a_{(n-1)n+2-(n-1)i} & \ldots & a_{(n-1)n+n-1-(n-1)i} \end{bmatrix}. \quad (4.9)$$

Proposition 4.5. Let $\alpha,\beta \in \{0, 1\}^n$ such that $\alpha \in G_n(\beta)$. Let B be a nonsingular matrix, $B = C^i(a_1, a_2, \ldots, a_n)$, where n and i are coprime and $(a_1, a_2, \ldots, a_n) \in \{0, 1\}^n$. Then $\alpha B \in G_n(\beta B)$.

Proof. As $\alpha \in G(\beta)$, then $\alpha = \rho^k(\beta)$, for some k such that $1 \leq k < n$. It is also clear that the columns $B_1, B_2, \ldots, B_n$ of the matrix $B = C^i(a_1, a_2, \ldots, a_n)$ are i-cyclic shift of each other, precisely, $B_j = \rho^{(j-1)i}B_1$. Now,

$$\begin{aligned} \beta B &= (\beta B_1, \beta B_2, \beta B_3, \ldots, \beta B_n), \\ &= (\beta B_1, \beta\rho^i(B_1), \beta\rho^{2i}(B_1), \ldots, \beta\rho^{(n-1)i}(B_1)), \\ &= (\beta B_1, \rho^{n-i}(\beta)B_1, \rho^{n-2i}(\beta)B_1, \ldots, \rho^i(\beta)B_1). \end{aligned} \quad (4.10)$$

Again,

$$\begin{aligned} \alpha B &= (\alpha B_1, \alpha B_2, \alpha B_3, \ldots, \alpha B_n), \\ &= (\alpha B_1, \rho^{n-i}(\alpha)B_1, \rho^{n-2i}(\alpha)B_1, \ldots, \rho^i(\alpha)B_1), \\ &= (\rho^k(\beta)B_1, \rho^{n-i}(\rho^k(\beta))B_1, \rho^{n-2i}(\rho^k(\beta))B_1, \ldots, \rho^i(\rho^k(\beta))B_1), \\ &= (\rho^k(\beta)B_1, \rho^{n-i+k}(\beta)B_1, \rho^{n-2i+k}(\beta)B_1, \ldots, \rho^{i+k}(\beta)B_1). \end{aligned} \quad (4.11)$$

Since i and n are coprime, for some integer γ we have, $\gamma i \equiv 1 \bmod n$, i.e., $\gamma k i \equiv k \bmod n$, i.e., $ri \equiv k \bmod n$, as $\gamma k \equiv r \bmod n$, for some r, $0 \leq r < n$. Therefore, in the expression of αB, we have, $\rho^{(n-ri+k)}(\beta)B_1 = \beta B_1$, $\rho^{(n-(r+1)i+k)}(\beta)B_1 = \rho^{(n-i)}(\beta)B_1$ and in this way all the elements of $\{\beta B_1, \rho^{n-i}(\beta)B_1, \rho^{n-2i}(\beta)B_1, \ldots, \rho^{i}(\beta)B_1\}$ will appear in αB in the same sequence in which they occur in βB. If τ be the term of βB, which occurs as the n-th term of αB, then all the remaining terms of βB after τ will appear in the same sequence starting from the 1st position up to the $(r-2)$-th position in αB. Therefore $\alpha B \in G_n(\beta B)$. Hence the proof.

Similar to the Proposition 4.4, using Proposition 4.5 we get the following.

Proposition 4.6. Let $f(x)$ be an n-variable RSBF and B be an $n \times n$ nonsingular binary matrix as explained in Proposition 4.5. Then $f(xB)$ is also an RSBF.

In our case, $n = 9$ and we choose $i = 2$. As for example, one may consider the matrix

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{4.12}$$

It is found that, exploiting this matrix, the nine RSBFs can be represented by two RSBFs which are different up to the affine equivalence. Note that the autocorrelation spectra of these two RSBFs are different (as can be seen from Table 4.3), hence there is no affine equivalence between them. Below these two representative RSBFs having nonlinearity 241 are presented, the first one with absolute indicator 52 and the second one with absolute indicator 76.

```
05777A7A6ED82E887CFCE3C549E994947AE4FBA5B91FE46674C3AC8386609671
3FCCAC20EE9B9966CAD357AAE921286D7A20A55A8DF0910BC03C3C51866D2B16
```

```
04757A727ED96F087EFCE2C768EB04947AECFBA5B91DE42E7CC1AC8B1060D671
2FCCEDB0EE8B8926CAD357A2E92148ED3AB4A1128DF0918B46143C51A66D2B16
```

4.4 Conclusions

In this chapter, 9-variable RSBFs having nonlinearity > 240 are enumerated by an efficient exhaustive search strategy. It is found that there are 8×189 many RSBFs having nonlinearity 241 and there is no RSBF having nonlinearity > 241. On the other hand, exploiting binary nonsingular circulant matrices and some variants of them, it is shown that there are only two different 9-variable RSBFs having nonlinearity 241 up to the affine equivalence.

CHAPTER 5

NEW CLASSES OF BOOLEAN FUNCTIONS

5.1 Introduction

As the space of the RSBF class is much smaller ($\approx 2^{\frac{2^n}{n}}$) than the total space of Boolean functions (2^{2^n}) on n variables, it is possible to exhaustively search the space of RSBFs up to a certain value of n. In [24], an exhaustive search carried out for the whole space (2^{60}) of 9-variable RSBFs exploiting some combinatorial results related to the Walsh spectra of RSBFs, has shown that there is no RSBF having nonlinearity greater than 241. Consequently, in order to find Boolean functions with higher nonlinearity, one needs to increase the search space.

Motivated by this fact, we have firstly proposed the generalized k-RSBFs, as functions which satisfy $f(\alpha^{2^k}) = f(\alpha)$, where $1 \leq k \mid n$. Note that if $k = 1$, the resulting functions are the same as idempotents; whereas for $k = n$ the entire space of n-variable Boolean functions is covered. In the space of generalized k-RSBFs, imposing the condition of invariance under the action of dihedral group, we have defined the class of generalized k-DSBFs as a subset of k-RSBFs.

Secondly, we have used the steepest-descent-like iterative algorithm in [25] for a search in the generalized 3-DSBF and 3-RSBF classes. This search has successfully ended up with 9-variable Boolean functions in both of these classes, having nonlinearity 242, absolute indicator values 32, 40 and 56. This result shows that the covering radius of the first order Reed-Muller code $R(1, 9)$ is at least 242. This result is also important for $n = 11$ and $n = 13$, since the bent concatenation of 9-variable functions with nonlinearity 242 leads to the construction of 11-variable and 13-variable functions with nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 2\times 2^{\frac{n-9}{2}})$, which exceeds the bent concatenation bound by $2\times 2^{\frac{n-9}{2}}$ (see Table 5.1). However, we should mention that for odd $n \geq 15$, the nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^{\frac{n-15}{2}})$ given in [52] that can be obtained by concatenating 15-variable Patterson-Wiedemann functions is still greater than the nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 2 \times 2^{\frac{n-9}{2}})$. In Table 5.1, we present the bent concatenation bound for $7 \leq n \leq 15$, together with recent nonlinearity results.

Table 5.1: Summary of Nonlinearity Results for $n = 7, 9, 11, 13, 15$

n	7	9	11	13	15
Bent Concatenation Bound: $2^{n-1} - 2^{\frac{n-1}{2}}$	56	240	992	4032	16256
Nonlinearity Results in Chapter 4	–	241	994	4036	16264
Nonlinearity Results in this chapter	–	242	996	4040	16272
Patterson-Wiedemann Construction [52]	–	–	–	–	16276
Upper Bound [23]	56	244	1000	4050	16292

Thirdly, knowing the fact that RSBFs are invariant under a special type of permutation, we have investigated the same search problem from a different direction and considered the possibility of other ‘*rich*’ classes that are invariant under some permutations. Linearly equivalent Boolean functions have the same nonlinearity; therefore, while searching for highly nonlinear functions, it is quite logical to classify all $n!$ permutations up to the linear equivalence of Boolean functions that are invariant under them. More specifically, for 9-variable functions, we have classified 9! many permutations into 30 classes which are different up to the linear equivalence of functions that are invariant under them. Then for each class, by picking up a representative permutation arbitrarily, we have searched the corresponding set of functions. Consequently, in some of these sets, we have obtained 9-variable functions with nonlinearity 242 and absolute indicator values 40, 48 & 56. So, our aim of defining other ‘*rich*’ classes is accomplished. However, the functions presented in this chapter do not contain any zero in their Walsh spectra, and hence, they cannot be linearly transformed to balanced functions.

In the following section, we introduce the generalized rotation symmetric and dihedral symmetric Boolean functions. Classification of permutations on inputs of 9-variable Boolean functions, with respect to the linear equivalence of Boolean functions that are invariant under them, is presented in Section 5.5. Different results related to 9-variable Boolean functions with nonlinearity 242 are presented in both Section 5.3 and Section 5.5. Finally, some additional 11 and 13-variable DSBFs, which are attained by the steepest-descent-like search algorithm with nonlinearities 994 and 4036 respectively, are presented in Section 5.4. It should be noticed that those functions have exactly the same nonlinearities, as those would be obtained by concatenating 9-variable functions with nonlinearity 241.

5.2 Generalized k-RSBFs and k-DSBFs

After recalling RSBFs, we propose the generalized classes of k-RSBFs and k-DSBFs in Definition 5.1 and Definition 5.2 respectively. Letting $(x_0, x_1, \ldots, x_{n-1}) \in \{0,1\}^n$, the (left) k-cyclic shift operator ${\rho^k}_n$ on n-tuples is defined as

$$ {\rho^k}_n(x_0, x_1, \ldots, x_{n-1}) = (x_{(0+k) \bmod n}, \ldots, x_{(n-1+k) \bmod n}), \tag{5.1}$$

for $1 \leq k \leq n$.

A Boolean function f is called *rotation symmetric* if for each input $(x_0, \ldots, x_{n-1}) \in \{0,1\}^n$, $f({\rho^1}_n(x_0, \ldots, x_{n-1})) = f(x_0, \ldots, x_{n-1})$. An orbit generated by $(x_0, x_1, \ldots, x_{n-1})$ is denoted by $G_n(x_0, x_1, \ldots, x_{n-1}) = \{{\rho^k}_n(x_0, x_1, \ldots, x_{n-1}) \mid 1 \leq k \leq n\}$ and the number of such orbits is g_n ($\approx 2^{\frac{2^n}{n}}$). More specifically, g_n is equal to $\frac{1}{n}\sum_{t|n} \phi(t) 2^{\frac{n}{t}}$ [67], where $\phi(t)$ is the Euler's phi-function. The total number of n-variable RSBFs is 2^{g_n}.

In the following, we define the generalized RSBFs as k-rotation symmetric Boolean functions (k-RSBFs).

Definition 5.1. Let $1 \leq k \leq n, k \mid n$. An n-variable Boolean function f is called *k-rotation symmetric* if for each input $(x_0, \ldots, x_{n-1}) \in \{0,1\}^n$, $f({\rho^k}_n(x_0, \ldots, x_{n-1})) = f(x_0, \ldots, x_{n-1})$.

As can be seen, the k-rotation symmetric Boolean functions are invariant under k-cyclic rotations of inputs. Therefore, an orbit of a k-RSBF generated by $(x_0, x_1, \ldots, x_{n-1})$ is ${G^k}_n(x_0, x_1, \ldots, x_{n-1}) = \{{\rho^i}_n(x_0, x_1, \ldots, x_{n-1}) \mid i = k, 2k, 3k, \ldots, n\}$. For example, ${G^3}_9(001\ \ 001\ 111) = \{(001\ 001\ 111), (001\ 111\ 001), (111\ 001\ 001)\}$.

If $g_{n,k}$ is the number of distinct orbits in the class of k-RSBFs of n variables, one can show that $g_{n,k} = \frac{k}{n}\sum_{t|\frac{n}{k}} \phi(t) 2^{\frac{n}{t}} (\approx 2^{k \times \frac{2^n}{n}})$, where $\phi(t)$ is the Euler's phi function.

In [40], a subspace of RSBFs called Dihedral Symmetric Boolean Functions (DSBFs), which are invariant under the action of dihedral group D_n are introduced. In addition to the (left) k-cyclic shift operator ${\rho^k}_n$ on n-tuples, which is defined previously, the dihedral group D_n also includes the reflection operator $\tau_n(x_0, x_1, \ldots, x_{n-1}) = (x_{n-1}, \ldots, x_1, x_0)$. The $2n$ permutations of D_n are then defined as $\{{\rho^1}_n, {\rho^2}_n, \ldots, {\rho^{n-1}}_n, {\rho^n}_n, {\tau^1}_n, {\tau^2}_n, \ldots, {\tau^{n-1}}_n, {\tau^n}_n\}$. The dihedral group D_n generates equivalence classes in the set $\{0, 1\}^n$ [55]. Let d_n be the number of such partitions. The following proposition gives the exact count of d_n [19, page 184], [40].

Proposition 5.1. Let d_n be the total number of orbits induced by the dihedral group D_n acting on $\{0, 1\}^n$. Then $d_n = g_n/2 + l$, where, $g_n = \frac{1}{n}\sum_{t|n}\phi(t)2^{\frac{n}{t}}$ is the number of rotation symmetric classes [67], $\phi(t)$ is the Euler's phi-function and

$$l = \begin{cases} \frac{3}{4}2^{\frac{n}{2}}, & \text{if } n \text{ is even,} \\ 2^{\frac{n-1}{2}}, & \text{if } n \text{ is odd.} \end{cases} \tag{5.2}$$

Since there are 2^{d_n} many n-variable DSBFs and $d_n \approx 2^{\frac{2^n}{2n}}$, a reduction in the size of the search space over the size of RSBFs is provided.

Definition 5.2. Let $1 \leq k \leq n$, $k\,|\,n$. An n-variable Boolean function f is called *k-dihedral symmetric* if f is invariant under the group action $D^k{}_n = \{\rho^i{}_n, \tau_n\rho^i{}_n \,|\, i = k, 2k, 3k, ..., n\}$.

As the class of DSBFs is a subspace of k-DSBFs, we call k-DSBFs "generalized dihedral symmetric Boolean functions". One should observe that k-DSBFs is a subspace of k-RSBFs.

When Proposition 5.1 is applied to k-dihedral symmetric functions, we obtain the following corollary.

Corollary 5.1. Let $d_{n,k}$ be the number of distinct orbits, in the class of k-DSBFs of n variables. Then, $d_{n,k} = g_{n,k}/2 + l$, where, $g_{n,k} = \frac{k}{n}\sum_{t|\frac{n}{k}}\phi(t)2^{\frac{n}{t}}$ is the number of k-rotation symmetric classes, $\phi(t)$ is the Euler's phi-function and

$$l = \begin{cases} 2^{\frac{n}{2}-1}, & \text{if } n \text{ is even, } k \text{ is even,} \\ \frac{3}{4}2^{\frac{n}{2}}, & \text{if } n \text{ is even, } k \text{ is odd,} \\ 2^{\frac{n-1}{2}}, & \text{if } n \text{ is odd.} \end{cases} \tag{5.3}$$

Table 5.2 compares the orbit counts of k-rotational classes, k-dihedral classes, RSBFs, and DSBFs for $k|n, n \leq 15$.

Table 5.2: Comparison of the orbit counts $g_n, d_n, g_{n,k}$ and $d_{n,k}$ (for $n = 4, 6, \ldots, 15$, and all integers k, which divide n).

n		k	2	3	4	5	6	7
4	$g_4 = 6$	$g_{4,k}$	10	–	–	–	–	–
	$d_4 = 6$	$d_{4,k}$	7	–	–	–	–	–
6	$g_6 = 14$	$g_{6,k}$	24	36	–	–	–	–
	$d_6 = 13$	$d_{6,k}$	16	24	–	–	–	–
8	$g_8 = 36$	$g_{8,k}$	70	–	136	–	–	–
	$d_8 = 30$	$d_{8,k}$	43	–	76	–	–	–
9	$g_9 = 60$	$g_{9,k}$	–	176	–	–	–	–
	$d_9 = 46$	$d_{9,k}$	–	104	–	–	–	–
10	$g_{10} = 108$	$g_{10,k}$	208	–	–	528	–	–
	$d_{10} = 78$	$d_{10,k}$	120	–	–	288	–	–
12	$g_{12} = 352$	$g_{12,k}$	700	1044	1376	–	2080	–
	$d_{12} = 224$	$d_{12,k}$	382	570	720	–	1072	–
14	$g_{14} = 1182$	$g_{14,k}$	2344	–	–	–	–	8256
	$d_{14} = 687$	$d_{14,k}$	1236	–	–	–	–	4224
15	$g_{15} = 2192$	$g_{15,k}$	–	6560	–	10944	–	–
	$d_{15} = 1224$	$d_{15,k}$	–	3408	–	5600	–	–

5.3 9-variable 3-DSBFs and 3-RSBFs

We apply our search strategy to 9-variable 3-DSBFs, where the size of the 3-DSBF search space is 2^{104} (see Table 5.2). We have found several unbalanced Boolean functions having nonlinearity 242. Among them there are two different absolute indicator values, which are 32 and 40. The following is the truth table of a 9-variable, 3-dihedral symmetric Boolean function having nonlinearity 242, absolute indicator value 40, and algebraic degree 7:

```
68B7EF2DA03B0D3EA00DB6A96DD99AEAFDB9C842B6D5DC8C4526CE0DD29020DB
B75FE3314568344E73688FF0CB2482E065231869E1AA4583765CC491F8A8DB12
```

And, the function below is another 9-variable 3-DSBF having nonlinearity 242, absolute indicator value 32, and algebraic degree 7:

```
125425D30A398F36508C06817BEE122E250D973314F976AED58A3EA9120DA4FE
0E4D4575C42DD0426365EBA7FC5F45BE9B2F336981B5E1863618F49474F6FE00
```

We present the C code of our search algorithm in Appendix D. Using a computer system with Pentium IV 2.8 GHz processor and 256 MB RAM having Windows XP operating system, and setting the maximum iteration number to $N = 60,000$, a typical run of the search algorithm takes 1 minute and 17 seconds. We have carried out 100 runs, each with $N = 60,000$. Out of 6 million distinct 3-DSBFs, 152 functions have the nonlinearity 241, and 36 many 3-DSBFs have the nonlinearity 242.

Additionally, we have applied the search strategy to 9-variable 3-RSBFs (the size of the search space is now 2^{176} as can be seen from Table 5.2), for which we initiate the search algorithm with a 9-variable 3-DSBF having nonlinearity 242. Then we have obtained some 9-variable 3-RSBFs (which are not in 3-DSBFs) having nonlinearity 242, absolute indicator 56, and algebraic degree 7. The following is the truth table of such a function:

```
3740B6A118A1E19642A85E2B7E2F3C3CB65FA0D95EC9DB1EA92BDB3666185AE0
087F5FE6E0757106A12FC918754C40E8A1BCCB7A714032A8961456E066E8A801
```

It is clear that using one of the above 9-variable Boolean functions (say f) and a 2-variable bent function (say g), the 11-variable function $g(y_0, y_1) \oplus f(x_0, \ldots, x_8)$ with highest -till date- nonlinearity of $2^{11-1} - 2^{\frac{11-1}{2}} + 4 = 996$, can be obtained. Similarly $h(y_0, y_1, y_2, y_3) \oplus f(x_0, \ldots, x_8)$ is the most nonlinear 13-variable function known to date, with nonlinearity $2^{13-1} - 2^{\frac{13-1}{2}} + 8 = 4040$ where h is a 4-variable bent function and f is one of the above 9-variable functions with nonlinearity 242. We think this is a significant improvement on the results of [25]. However, since the nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 2 \times 2^{\frac{n-9}{2}})$, which can be obtained by bent concatenation of 9-variable functions with nonlinearity 242 is less than the nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^{\frac{n-15}{2}})$ given in [52] for odd $n \geq 15$, this result is significant only for odd $13 \geq n \geq 9$.

5.3.1 Coding Theoretic Significance

The concept of *urcoset* was first presented in [22] and then in [4, 5] as *orphan coset*. The set D defines an urcoset, if the union of the support of the leaders of D covers the full space; in other words, a coset D of the first order Reed-Muller code $R(1, n)$ with a set of coset leaders $L(D)$ is an urcoset [32], when $\cup_{g \in L(D)} supp(g) = \{0, 1, \ldots, 2^n - 1\}$.

In [16], orphan cosets having minimum weight of 240 have been reported, and in [24] it is confirmed that each of the cosets $f \oplus R(1, 9)$ is an orphan or urcoset, where f is any RSBF having nonlinearity 241.

We have checked by running a computer program that for any of the above functions f having nonlinearity 242, each of the cosets $f \oplus R(1,9)$ is an orphan or urcoset. This is the first time orphan cosets having minimum weight 242 are demonstrated.

In [4], it is conjectured that the covering radius [33, 52] of $R(1,n)$ is even. Our results for $n = 9$ show that the covering radius is at least 242 and it is an interesting open question to settle it. The upper bound presented in [23, 21] for the covering radius of $R(1,9)$ is 244.

5.4 11 and 13-variable DSBFs

In [40], the class of Dihedral Symmetric Boolean Functions (DSBFs), a subset of the RSBF class, which is invariant under the action of the dihedral group, is introduced. It has been shown that some of the 9-variable RSBFs having nonlinearity 241 also belong to this subset, demonstrating the richness of DSBFs in terms of high nonlinearity. Motivated by this point, we have carried out a systematic search in the DSBF class for $15 >$ odd $n > 9$, and found Boolean functions having nonlinearity $> (2^{n-1} - 2^{\frac{n-1}{2}})$. More specifically, for 11-variable DSBFs, we have attained an 11-variable DSBF with nonlinearity 994 within the space of size 2^{126}. For 13-variable DSBFs, in order to reduce the search space (2^{380}), we have applied some additional permutations on input vectors, and obtained a subset of size 2^{74}, in which we have found several 13-variable DSBFs with nonlinearity 4036. Consequently, our trials confirm that the DSBF class contains highly nonlinear Boolean functions and it is a rich subset of the RSBF class for $n = 11, 13$, as well. We should also mention that, this is the first demonstration of Boolean functions on odd number of input variables $9 < n < 15$ having nonlinearity greater than the bent concatenation bound, which are not obtained by the bent concatenation of 9-variable Boolean functions with nonlinearity > 240.

For the 11-variable DSBF case for which the size of search space is 2^{126}, we have carried out 8000 runs of the search algorithm, and found an 11-variable DSBF having nonlinearity 994, absolute indicator value 200, and algebraic degree 9, which is given Appendix A. A typical run of the search algorithm takes 1 minute and 16 seconds using the same computer system.

For the 13-variable DSBF case, since its search space is huge (2^{380}), before starting the search we apply the following permutation in addition to the permutations of dihedral group on input vectors

$$\pi(x_0, x_1, \ldots, x_{12}) = (x_0, x_2, x_4, x_6, x_8, x_{10}, x_{12}, x_1, x_3, x_5, x_7, x_9, x_{11}) \tag{5.4}$$

such that for each input $(x_0, \ldots, x_{12}) \in \{0,1\}^{13}$,

$$f({\rho^1}_n(x_0, \ldots, x_{12})) = f(\tau_n(x_0, \ldots, x_{12})) = f(\pi(x_0, \ldots, x_{12})) = f(x_0, \ldots, x_{12}), \tag{5.5}$$

and the search space of 13-variable DSBFs is reduced from 2^{380} to 2^{74}. Note that this permutation constitutes a subset of 13-variable DSBFs for which, using similar combinatorial methods as in [24], it may be possible to carry out an exhaustive search to enumerate 13-variable DSBFs with nonlinearity $\geq$ 4036, with a reasonable amount of computational power. We have carried out 500 runs of the search algorithm, and found two 13-variable DSBFs having nonlinearity 4036 in this subset. One of them with nonlinearity 4036, absolute indicator value 208, and algebraic degree 10 is given in Appendix A. In this case, a typical run takes one minute using the same computer system.

Since these results confirm that the DSBF class contains highly nonlinear Boolean functions on 11 and 13-variables as well, it would be an interesting and open problem to attain some rich subsets achieving higher nonlinearity in the DSBF class.

5.5 Classification of Permutations

As it is deduced from the discussion in the preceding section, RSBFs are invariant under a special type of permutation. To search for better cryptographic characteristics, we consider the possibility of other classes of Boolean functions that are invariant under some permutations. Since linearly equivalent functions have the same nonlinearity, it makes sense to classify all $n!$ permutations up to the linear equivalence of Boolean functions that are invariant under them. The classification is based on the following proposition, which is easy to prove.

Proposition 5.2. Let f and g be Boolean functions which are invariant under arbitrary permutations π_f and π_g respectively. Then, f and g are said to be *linearly equivalent* if there exists a bijective linear mapping $L : \{0,1\}^n \to \{0,1\}$ such that $\pi_f = (L^{-1} \circ \pi_g \circ L)$.

Proof. Suppose $f(x) = g(L(x))$, i.e., $f = g \circ L$. Then, it holds that

$$f = g \circ L = g \circ \pi_g \circ L = f \circ L^{-1} \circ \pi_g \circ L = f \circ \pi_f. \tag{5.6}$$

Thus, we classify all possible permutations up to the equivalence

$$\pi_f \sim \pi_g \Leftrightarrow \exists L \text{ such that } \pi_f = (L^{-1} \circ \pi_g \circ L). \tag{5.7}$$

The classification can be accomplished through a computer program by exploiting the Jordan Normal Form for matrices. Specifically for 9-variable Boolean functions, all permutations of the identity matrix (362, 880 many) yield that there are only 30 permutations (see Table 5.3), which are different up to the equivalence defined above. Then, we apply the search algorithm for each class using its representative permutation and determine the corresponding nonlinearity given in the last column of Table 5.3. Our results show the existence of permutations having similar cryptographic characteristics with *k*-RSBFs and *k*-DSBFs.

From Table 5.3, it is seen that we have attained several 9-variable Boolean functions with nonlinearity 242, which we initially found in 3-DSBFs and 3-RSBFs, in the classes with sizes 2^{100}, 2^{104}, 2^{140}. In the following, we present 9-variable Boolean functions having nonlinearity 242 and different autocorrelation spectra from those of the functions found in 3-DSBFs and 3-RSBFs.

We have applied 100 runs of the search algorithm to the space of size 2^{104} and found two 9-variable Boolean functions with nonlinearity 242, absolute indicator value 48, and algebraic degree 7. A typical run takes the same amount of time as for the case of 3-DSBFs (since the sizes of both spaces are the same). One of these functions is given below:

```
7B8F94BAD364DAC9931906F9465FF33E921E13D7552DAFD684757B662FDA3C68
FA8D94B3C3659B5FCC46FD1518050F97A1E02039AAF74337134F30AB5B41D9DE
```

which is invariant under the representative permutation

$$\pi(x_0, x_1, \ldots, x_8) = (x_0, x_2, x_1, x_4, x_5, x_6, x_7, x_8, x_3). \tag{5.8}$$

Table 5.3: Classification of all possible 362,880 many permutations for 9-variable Boolean functions, and the best achieved nonlinearity result for each class.

Representative permutation	Number of permutations	Maximum number of input vectors in an orbit	Total number of distinct orbits	Best achieved nonlinearity result
(0,1,2,3,4,5,6,7,8) (*identity*)	1	1	512	239
(5,7,4,8,2,0,6,1,3)	945	2	272	240[6]
(3,1,7,0,5,4,6,2,8)	1260	2	288	240
(7,1,2,3,5,4,6,0,8)	378	2	320	240
(0,8,2,3,4,5,6,7,1)	36	2	384	240
(4,6,7,2,8,1,5,3,0)	**2240**	**3**	**176**	**242**[3,4]
(5,1,2,4,7,8,6,3,0)	3360	3	192	240
(0,1,2,8,4,5,6,3,7)	168	3	256	240
(1,4,7,5,6,2,0,3,8)	**11340**	**4**	**140**	**242**[2]
(4,7,5,6,0,1,3,2,8)	11340	4	168	240
(0,2,1,7,4,3,5,6,8)	7560	4	176	240
(0,8,2,3,4,1,6,5,7)	756	4	192	240
(0,7,2,5,4,1,3,6,8)	3024	5	128	239
(8,7,3,0,1,6,2,4,5)	**20160**	**6**	**100**	**242**[1]
(0,2,1,4,5,6,7,8,3)	**30240**	**6**	**104**	**242**
(7,1,0,3,4,2,8,6,5)	10080	6	112	240
(7,4,0,5,1,8,6,2,3)	10080	6	144	240
(8,4,3,2,1,7,5,6,0)	2520	6	144	240
(0,6,2,7,8,1,5,3,4)	7560	6	160	240
(8,1,3,2,4,5,0,7,6)	2520	6	192	240
(2,0,6,1,4,5,7,8,3)	25920	7	80	240
(0,3,5,8,1,4,7,2,6)	45360	8	72	240
(1,6,7,4,8,2,5,3,0)	**40320**	**9**	**60**	**241**[5]
(5,8,6,7,2,0,1,3,4)	9072	10	80	240
(1,3,8,7,4,0,6,5,2)	18144	10	96	240
(3,5,7,1,6,0,8,2,4)	15120	12	88	240
(0,2,7,8,4,6,3,1,5)	15120	12	96	240
(4,5,7,1,0,8,3,6,2)	25920	14	60	240
(6,5,1,4,7,2,3,0,8)	24192	15	64	238
(4,8,1,2,6,7,5,0,3)	18144	20	48	240

[1] Nonlinearity result of 242 is attained in the subset of size 2^{74} in the set of size 2^{100}.
[2] Nonlinearity result of 242 is attained in the subset of size 2^{86} in the set of size 2^{140}.
[3] Nonlinearity result of 242 is attained in the subset of size 2^{104} in the set of size 2^{176}.
[4] The class contains the permutation corresponding to 3-RSBFs.
[5] The class contains the permutation corresponding to RSBFs.
[6] The class contains the permutation corresponding to 9-DSBFs.

Then, in order to reduce the search space, we have considered some subclasses. For this purpose, we have applied the reflection operator, which is defined as $\tau_n(x_0, x_1, \ldots, x_8) = (x_8, \ldots, x_1, x_0)$ for 9-variable Boolean functions, in addition to the representative permutation. As a result of this method, we have identified a subset of size 2^{74} in the set of size 2^{100}. In this subset, we have attained several 9-variable Boolean functions with nonlinearity 242, absolute indicators 40 and 64, and algebraic degree 7. One of them having absolute indicator 64 is provided below:

```
0331786B34D878855663A2E961F1CB4F779EBBF6881ABB24AC033E6C2B32E049
3D0891DB1888EA5E6F910310311532FC68D5F2A4B5BE6445E41F64299F0CC99A
```

which is invariant under the permutations of the reflection operator τ_n and the representative permutation

$$\pi(x_0, x_1, \ldots, x_8) = (x_8, x_7, x_3, x_0, x_1, x_6, x_2, x_4, x_5). \tag{5.9}$$

For this case, we have carried out 100 runs of the search algorithm resulting in 9 many Boolean functions with nonlinearity 242 such that seven of them with absolute indicator value of 64, and the remaining with that of 40. A typical run takes 1 minute and 4 seconds using the same computer system.

5.6 Conclusions

By suitably generalizing the class of RSBFs, we have introduced k-RSBFs, as functions which satisfy $f(\alpha^{2^k}) = f(\alpha)$, where the nonzero positive integer k divides n, and $\alpha \in GF(2^n)$. We have also defined the class of k-DSBFs as a subset of k-RSBFs imposing the condition of invariance under the action of dihedral group. Using the steepest-descent-like iterative algorithm in [25, 26] for a search in the generalized 3-DSBF and 3-RSBF classes, we have attained 9-variable 3-RSBFs and 3-DSBFs with nonlinearity 242. This result shows that the covering radius of the first order Reed-Muller code $R(1, 9)$ is at least 242 and there exist Boolean functions on n variables having nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 2 \times 2^{\frac{n-9}{2}})$ for $n = 9, 11, 13$.

Further, we have considered the invariance of Boolean functions under all possible permutations which are classified up to the linear equivalence of Boolean functions that are invariant under them. Specifically for $n = 9$, there are 30 such classes. Exploiting the same search algorithm [25], we have attained 9-variable Boolean functions having nonlinearity 242 in the classes with sizes 2^{104} and 2^{140}. Then, we have considered some subclasses by adding permutation of the reflection operator τ_n to the representative permutation. As a result, we have identified a subset of size 2^{74}, in the set of size 2^{100}, having 9-variable Boolean functions with nonlinearity 242. Considering the combinatorial search techniques in [24], we note that it may be possible to exhaustively search the subset of size 2^{74} for the enumeration of 9-variable Boolean functions having nonlinearity ≥ 242, with a reasonable amount of computational power. Moreover, we have obtained an 11-variable DSBF having nonlinearity 994 and several 13-variable DSBFs having nonlinearity 4036, which confirm the richness of DSBFs [40] in terms of high nonlinearity for $n = 11$ and 13.

We think that the results that we present contain significant information on the existence of maximum nonlinearity-Boolean functions with odd number of input variables, within the classes that are invariant under some permutations.

CHAPTER 6

BALANCED BOOLEAN FUNCTIONS ON 13-VARIABLES

6.1 Introduction

As balancedness is an important cryptographic and combinatorial property of Boolean functions, balanced Boolean functions with odd number of variables having nonlinearity greater than the bent concatenation bound have received lot of attention in the literature. In fact, in the literature, such functions could be constructed exploiting Boolean functions having nonlinearity greater than bent concatenation bound. Below the existing results in this area are listed (we will refer 15-variable Boolean functions having nonlinearity 16276 as PW functions since these function were found by Patterson and Wiedemann).

1. In [62], balanced Boolean functions having nonlinearity greater than bent concatenation bound of $(2^{n-1} - 2^{\frac{n-1}{2}})$ could be found for odd $n \geq 29$, using the PW functions as black box.
2. In [59, 39], 15-variable balanced Boolean functions having nonlinearity $2^{15-1} - 2^{\frac{15-1}{2}} + 6 = 16262$ have been constructed by modifying the structure of the PW functions with heuristic search.
3. In [61], modifying the structure of the PW functions systematically in the space of rotation symmetric Boolean functions, 15-variable Boolean function having nonlinearity $2^{15-1} - 2^{\frac{15-1}{2}} + 10 = 16272$ has been constructed.
4. In [34], using the Boolean functions on 9-variables having nonlinearity 242, 13-variable balanced function having nonlinearity $2^{13-1} - 2^{\frac{13-1}{2}} + 2 = 4034$ has been constructed.

The Boolean functions on 9-variables having nonlinearity 242, presented in the previous chapter, do not contain any zero in the Walsh spectrum and therefore they cannot be linearly transformed to balanced functions. In [34], these functions are used to construct 13-variable balanced function having nonlinearity $2^{13-1} - 2^{\frac{13-1}{2}} + 2 = 4034$ which is the first demonstration of balanced Boolean functions on odd number of variables having nonlinearity strictly greater than the bent concatenation bound for number of input variables less than 15. We improve the search algorithm used in [34], and then arrive at 13-variable balanced functions having nonlinearity 4036.

6.2 13-variable Balanced Functions with Nonlinearity 4036

Sinilar to the idea of [59, 39], the strategy described in [34] is based upon the concept of searching balanced functions within the eight-bit neighborhood of an 13-variable unbalanced function F with $nl(F) = 4040$, and $W_F(0) = \pm 16$. The 13-variable unbalanced function is constructed from 9-variable unbalanced functions which are highly nonlinear. The minimum valued spectral components (± 4) of 9-variable unbalanced functions are translated to the origin; so that the resulting 13-variable function has $W_F(0) = \pm 16$ and it is therefore probable to find a highly nonlinear but balanced functions in its 8-bit neighborhood.

Once F is constructed, the problem is reduced to a search in a space of size $\binom{2^{12}+8}{8}$, which is approximately equal to 2^{81}. The search algorithm [34] used in finding the 13-variable balanced function with nonlinearity 4034 randomly toggles eight many positions of the truth table of F from 0 to 1. We have carried out 15 million trials of this algorithm, which takes 62 hours by using a computer system with Pentium IV 2.8 GHz processor and 256 MB RAM having Windows XP operating system. The search has resulted in 238 functions having nonlinearity 4034 and 14,999,762 many functions having nonlinearity 4032. The distribution of nonlinearities found by random search of this algorithm demonstrates the rareness of 13-variable balanced functions with nonlinearity greater than 4032.

In order to reduce the search time, we have then used our steepest-descent-like iterative algorithm, which has recently proved its effectiveness with the results for 9-variable functions of nonlinearity 241 [25] and 242 [30]. Calling the input of an iteration, F_{in}, each iteration step of the algorithm firstly computes the costs of all functions in a pre-defined neighborhood S of F_{in}. Then, the function with the smallest cost is chosen as the iteration output F_{out}, provided that it is different from the outputs of all previous iteration steps. S is not allowed to contain F_{in}; therefore, it is possible in some cases that the cost of F_{out} is larger than that of F_{in}. Recall that this is one of the critical parts of the search strategy, which provides its ability to escape from local minima. The proper choice of the cost function is also crucial.

We adapt our steepest-descent-like algorithm to the search of highly nonlinear and balanced 13-variable functions in the eight-bit neighborhood of F. The set S used in our algorithm is defined as the intersection of three sets: *i*) 13-variable balanced functions, *ii*) 8-bit neighborhood of a 13-variable function F with $nl(F) = 4040$, and $W_F(0) = \pm 16$, *iii*) 2-bit neighborhood of F_{in}, which changes at each iteration step. This intersection set contains 8×2^{12} many balanced functions.

The choice of a suitable cost function is very important. We base our choice upon the following intuition: Let the 2-bit neighborhood of an n-variable function f be partitioned by 3 sets,

$$A = \{\text{functions } f_a \text{ having the same nonlinearity } nl(f)\ as\ f\}, \tag{6.1a}$$

$$B = \{\text{functions } f_b \text{ with nonlinearity } (nl(f) - 2)\}, \tag{6.1b}$$

$$C = \{\text{functions } f_c \text{ with nonlinearity } (nl(f) + 2)\}. \tag{6.1c}$$

Denoting ${W_f}^{max}$ as the maximum magnitude in the Walsh spectrum of f and M_f as the number of spectral components with value $\pm {W_f}^{max}$; functions f_b with small M_{f_b} are much probable than those with large M_{f_b}. So, the cost function that we assign should ease the passage of the algorithm from Set B to Set A by favoring small values of M_f as follows:

$$Cost(f) = (2^{n+1} + M_f){W_f}^{max}. \tag{6.2}$$

The term ${W_f}^{max}$ in the cost expression is used to commend high nonlinearity, whereas M_f punishes large number of maximum magnitude-components in the Walsh spectrum, and the bias term $2^{n+1}{W_f}^{max}$ is necessary to direct the search in favor of functions f for which $Cost(f) > Cost(g)$ just because $M_f > M_g$; although $nl(f) > nl(g)$.

Setting the maximum iteration number to 500, a typical run of the search algorithm takes less than 3 hours using a computer system with Pentium IV 2.8 GHz processor and 256 MB RAM having Windows XP operating system, as above. Almost each step of the algorithm finds a balanced function with nonlinearity 4036, except for a minority with nonlinearity 4034 or 4032. Table 6.1 gives the nonlinearity distribution for three different runs, each starting with a different initial function F, which is constructed by using one of the three 9-variable functions of nonlinearity 242, as in the example described below. In Table 6.1, we use the notation $(n, nl, \triangle, d)$ to indicate the (*number of input variables*, *nonlinearity*, *absolute indicator*, *degree*) of the corresponding function.

Table 6.1: Nonlinearity distribution of 13-variable balanced functions found in a total number of 500 iterations of the steepest-descent-like algorithm.

Run #	$(n, nl, \triangle, d)$	4032	4034	4036
1	(9, 242, 32, 7)	1	92	407
2	(9, 242, 40, 7)	1	11	488
3	(9, 242, 56, 7)	3	6	491

An example for a balanced function with nonlinearity 4036 is presented in Appendix B, which is obtained by flipping eight bits of the initial function F described below. The toggled bits of F correspond to the indices 4667, 4758, 4807, 4823, 4913, 5042, 8133, 8187, where the truth table is indexed from 0 to 8191.

Details of the initial 13-variable function F of this example, having $nl(F) = 4040$ and $W_F(0) = 16$ are as follows: We utilize the unbalanced 9-variable function f with nonlinearity 242, absolute indicator 32 and degree 7, for which the corresponding truth table is given as follows:

```
125425D30A398F36508C06817BEE122E250D973314F976AED58A3EA9120DA4FE
0E4D4575C42DD0426365EBA7FC5F45BE9B2F336981B5E1863618F49474F6FE00
```

As in [34], we choose $w_1 = (0, 0, 0, 0, 1, 1, 0, 1, 1)$ so that the linear transformation $f_1(x) = f(x) \oplus w_1 \cdot x$, generates a function $f_1(x)$ with $W_{f_1}(0) = 4$; since the Walsh spectrum value of f corresponding to w_1 is equal to 4. We then construct the 13-variable function F by the direct sum of 9-variable function $f_1(x)$ with the 4-variable bent function $h(y_0, y_1, y_2, y_3)$ as

$$F = h(y_0, y_1, y_2, y_3) \oplus f_1(x_0, \ldots, x_8), \tag{6.3}$$

where $h = (0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0)$.

6.3 Conclusions

The strategy [34, 59, 39] that we have used is based upon the concept of searching balanced functions within the K-bit neighborhood of an unbalanced function with high nonlinearity, where the parameter $\pm 2K$ corresponds to the smallest value of its Walsh spectrum. The most elegant part of this concept is to translate the minimum valued spectral component ($\pm 2K$) of this unbalanced function to the origin so that $W_F(0) = \pm 2K$. Toggling K zeros or ones of F depending on the sign of $W_F(0)$, one obtains a balanced function with nonlinearity greater than or equal to $(nl(F) - K)$.

As in [34], to make $nl(F)$ as high as possible, we exploit the 9-variable unbalanced functions with nonlinearity 242. The 13-variable unbalanced functions obtained by bent concatenation have the nonlinearity 4040, and the smallest value in their Walsh spectra is ±16. Resulting balanced function of nonlinearity 4036 given in Appendix B improves the result in [34].

CHAPTER 7

CONCLUSIONS

In Chapter 3, as a major result, we show the existence of Boolean functions having nonlinearity $> (2^{n-1} - 2^{\frac{n-1}{2}})$ for $n = 9, 11, 13$, which remained as an open question in the literature for almost three decades. Further we attain balanced Boolean functions on $9, 10$ and 11 variables having maximum absolute value in the autocorrelation spectrum $< 2^{\lceil \frac{n}{2} \rceil}$. Earlier such Boolean functions were known for 15, 21 variables; on even number of variables, there was no evidence of such Boolean functions. Some of these Boolean functions on 9 and 11 variables can be affinely transformed to obtain first-order resiliency or first-order propagation characteristics. Some of these Boolean functions on 10-variable case can be affinely transformed to yield first order propagation characteristics. Moreover, we present a 10-variable Boolean function having first-order resiliency and nonlinearity 492, which was posed as an open question at Crypto 2000. The Boolean functions reported in this chapter are discovered using the steepest-descent-like iterative search algorithm [29, 25], an efficient search technique with an outstanding ability to escape from local optima, in the class of Rotation Symmetric Boolean Functions (RSBFs) along with proper affine transformations.

In Chapter 4, using the nice combinatorial structure of the Walsh spectra for RSBFs on odd number of variables [43], we efficiently perform the exhaustive search to enumerate 9-variable RSBFs having nonlinearity > 240. Consequently, we find that there is no RSBF achieving nonlinearity > 241 and there are 8×189 many RSBFs having nonlinearity 241. We further show that these RSBFs are represented by only two distinct RSBFs up to the affine equivalence. This result is obtained by utilizing some larger class of binary nonsingular circulant matrices.

Then, in Chapter 5, we improve the nonlinearity result of 241 to 242, which shows the existence of n-variable Boolean functions having nonlinearity $(2^{n-1} - 2^{\frac{n-1}{2}} + 2 \times 2^{\frac{n-9}{2}})$ for $n = 9, 11, 13$. This result is attained by suitably generalizing the classes of RSBFs and Dihedral Symmetric Boolean functions (DSBFs). More specifically, we introduce generalized RSBFs (k-RSBFs) as functions which satisfy $f(\alpha^{2^k}) = f(\alpha)$, where $1 \leq k \,|\, n$, $\alpha \in GF(2^n)$, and describe the class of k-DSBFs as a subset of k-RSBFs imposing the condition of invariance under the action of dihedral group. Further, we classify all possible permutations ($9! = 362,880$) on input variables of 9-variable Boolean functions up to the linear equivalence of Boolean functions that are invariant under some permutations and find that there are 30 such classes. Then, in some of these classes and their subsets, we identify new 9-variable Boolean functions having nonlinearity 242 with different autocorrelation spectra from those of the functions found in generalized RSBF and DSBF classes. In particular, we find a subset of size 2^{74} (in the class of size 2^{100}), much smaller than the sizes of 2^{176} and 2^{104} for the generalized RSBFs and DSBFs respectively, containining 9-variable Boolean functions having nonlinearity 242. Moreover, we obtain an 11-variable DSBF having nonlinearity 994 and several 13-variable DSBFs having nonlinearity 4036, which confirms the richness of DSBFs [40] in terms of high nonlinearity for $n = 11$ and 13. These functions are given in Appendix A. As in Chapter 3, we perform the steepest-descent-like iterative search algorithm to discover the Boolean functions presented in this chapter.

Finally, in Chapter 6, following the strategy used in [34, 59, 39], we modify the steepest-descent-like algorithm to attain balanced 13-variable Boolean functions having nonlinearity 4036, which improves the nonlinearity result of 4034 in [34]. Basically, the strategy is to searh balanced functions within the K-bit neighborhood of an unbalanced function with high nonlinearity, where the parameter $\pm 2K$ corresponds to the smallest value of its Walsh spectrum. As in [34], to construct such highly nonlinear unbalanced functions, we utilize the 9-variable unbalanced functions having nonlinearity 242 presented in Chapter 5. Resulting balanced 13-variable function having nonlinearity 4036 is given in Appendix B, which is identified within the 8-bit neighborhood of a 13-variable unbalanced function of nonlinearity 4040 obtained by bent concatenation.

At the end of this thesis, we point out some challenging open problems in the area:

1. Do there exist 8-variable balanced Boolean functions having nonlinearity 118?

2. In [14], Dobbertin has conjectured that the nonlinearity of a balanced Boolean function on n-variables cannot exceed $(2^{n-1} - 2^{\frac{n}{2}} + nl_{max}(f))$ where $nl_{max}(f)$ denote the maximum achievable nonlinearity of a balanced Boolean function f on $\frac{n}{2}$ variables. Can this conjecture be disproved?

3. In [4], it is conjectured that the covering radius [33, 52] of $R(1, n)$ is even. Our results for $n = 9$ show that the covering radius is at least 242. The upper bound presented in [23, 21] for the covering radius of $R(1, 9)$ is 244. What is the covering radius of $R(1, 9)$?

4. Do there exist balanced Boolean functions on $n = 9$, 11 variables having nonlinearity $> (2^{n-1} - 2^{\frac{n-1}{2}})$?

REFERENCES

[1] Special issue on particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3), June 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):203–207, January 1972.

[3] E. Biham and A. Shamir. *Differential cryptanalysis of the Data Encryption Standard*. Springer-Verlag, Berlin, 1993.

[4] R. A. Brualdi, N. Cai, and V. Pless. Orphan structure of the first order Reed-Muller codes. *Discrete Mathematics*, 102:239–247, 1992.

[5] R. A. Brualdi and V. S. Pless. Orphans of the first order Reed-Muller codes. *IEEE Transactions on Information Theory*, 36(2):399–401, 1990.

[6] A. Canteaut and Trabbia M. Improved fast correlation attacks using parity-check equations of weight 4 and 5. In *Eurocrypt 2000*, volume 1807 of *LNCS*, pages 573–588. Springer-Verlag, 2000.

[7] J. Clark, J. Jacob, S. Maitra, and P. Stanica. Almost Boolean functions: The design of Boolean functions by spectral inversion. *Computational Intelligence*, 20(3):450–462, 2004.

[8] J. Clark, J. Jacob, S. Stepney, S. Maitra, and W. Millan. Evolving Boolean functions satisfying multiple criteria. In *INDOCRYPT 2002*, volume 2551 of *LNCS*, pages 246–259. Springer-Verlag, 2002.

[9] J. A. Clark and J. L. Jacob. Two-stage optimization in the design of Boolean functions. In *ACISP 2000*, volume 1841 of *LNCS*, pages 242–254. Springer-Verlag, 2000.

[10] T. W. Cusick and P Stanica. Fast evaluation, weights and nonlinearity of rotation-symmetric functions. *Discrete Mathematics*, 258(1–3):289–301, 2002.

[11] D. K. Dalai, K. C. Gupta, and S. Maitra. Results on algebraic immunity for cryptographically significant Boolean functions. In *INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 92–106. Springer-Verlag, 2004.

[12] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[13] C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers*, volume 561 of *LNCS*. Springer-Verlag, 1991.

[14] H. Dobbertin. Construction of bent functions and balanced Boolean functions with high nonlinearity. In *Fast Software Encryption*, volume 1008 of *LNCS*, pages 61–74. Springer-Verlag, 1995.

[15] E. Filiol and C. Fontaine. Highly nonlinear balanced Boolean functions with a good correlation-immunity. In *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *LNCS*, pages 475–488. Springer-Verlag, 1998.

[16] C. Fontaine. On some cosets of the first-order Reed-Muller code with high minimum weight. *IEEE Transactions on Information Theory*, 45(4):1237–1243, 1999.

[17] S. Gangopadhyay, P.H. Keskar, and S. Maitra. Patterson-wiedemann functions revisited. *Discrete Mathematics*, 306:1540–1556, 2002. A special issue containing selected papers from "R. C. Bose Centenary Symposium on Discrete Mathematics and Applications", December 2002.

[18] X. Guo-Zhen and J. Massey. A spectral characterization of correlation immune combining functions. *IEEE Transactions on Information Theory*, 34(3):569–571, May 1988.

[19] F. Harary. *Graph Theory*. Addison-Wesley Publishing Company, 1972.

[20] M. Hell, A. Maximov, and S. Maitra. On efficient implementation of search strategy for rotation symmetric Boolean functions. In *Ninth International Workshop on Algebraic and Combinatoral Coding Theory, ACCT 2004*, Black Sea Coast, Bulgaria, June 19–25 2004.

[21] T. Helleseth, T. Klove, and J. Mykkeltveit. On the covering radius of binary codes. *IEEE Transactions on Information Theory*, 24:627–628, September 1978.

[22] T. Helleseth and H. F. Mattson Jr. On the cosets of the simplex code. *Discrete Mathematics*, 56:169–189, 1985.

[23] X.-d. Hou. On the norm and covering radius of the first order Reed-Muller codes. *IEEE Transactions on Information Theory*, 43(3):1025–1027, 1997.

[24] S. Kavut, S. Maitra, S. Sarkar, and M. D. Yücel. Enumeration of 9-variable rotation symmetric Boolean functions having nonlinearity > 240. In R. Barua and T. Lange, editors, *INDOCRYPT 2006*, volume 4329 of *LNCS*, pages 266–279, Springer-Verlag, India, December, 2006.

[25] S. Kavut, S. Maitra, and M. D. Yücel. Search for Boolean functions with excellent profiles in the rotation symmetric class. *IEEE Transactions on Information Theory*, 53(5):1743–1751, May 2007. An earlier version of this paper is available under the title "There exist Boolean functions on n (odd) variables having nonlinearity $> 2^{n-1} - 2^{\frac{n-1}{2}}$ if and only if $n > 7$" at IACR eprint server, http://eprint.iacr.org/2006/181, May 28, 2006, last accessed date: August 2008.

[26] S. Kavut, S. Maitra, and M. D. Yücel. Autocorrelation spectra of balanced Boolean functions on odd number input variables with maximum absolute value $< 2^{\frac{n+1}{2}}$. In J-F. Michon, P. Valarcher, and J-B. Yunès, editors, *Boolean Functions: Cryptography and Applications – BFCA 2006*, pages 73–86, University of Rouen, France, March, 2006.

[27] S. Kavut and M. D. Yücel. Random permutations on input vectors of Boolean functions. In O. Masnyk-Hansen, J-F. Michon, P. Valarcher, and J-B. Yunès, editors, *Boolean Functions: Cryptography and Applications – BFCA 2008*, Copenhagen, Denmark, May 2008.

[28] S. Kavut and M. D. Yücel. Improved cost function in the design of Boolean functions satisfying multiple criteria. In *Indocrypt 2003*, volume 2904 of *LNCS*, pages 121–134. Springer-Verlag, 2003.

[29] S. Kavut and M. D. Yücel. A new algorithm for the design of strong Boolean functions (in turkish). In *First National Cryptology Symposium*, pages 95–105, METU, Ankara, Türkiye, November 18–20 2005.

[30] S. Kavut and M. D. Yücel. Generalized rotation symmetric and dihedral symmetric Boolean functions – 9 variable Boolean functions with nonlinearity 242. In *17th International Applied Algebra, Algebraic Algorithms, and Error Correcting Codes (AAECC–17) Symposium*, volume 4851 of *LNCS*, pages 321–329, Springer-Verlag, Bangalore, India, December 2007.

[31] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[32] P. Langevin. On the orphans and covering radius of the Reed-Muller codes. In H. F. Mattson, T. R. N. Rao, and T. Mora, editors, *AAECC–9*, volume 539 of *LNCS*, pages 234–240. Springer-Verlag, 1991.

[33] F. J. MacWillams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.

[34] S. Maitra. Balanced Boolean function on 13-variables having nonlinearity strictly greater than the bent concatenation bound. Available at IACR eprint server, http://eprint.iacr.org/2007/309.pdf, May 2007, last accessed date: August 2008.

[35] S. Maitra. Highly nonlinear balanced Boolean functions with very good autocorrelation property. In *Workshop on Coding and Cryptography - WCC 2001*, volume 6 of *Electronic Notes in Discrete Mathematics*, pages 481–490, Elsevier, 2001.

[36] S. Maitra, S. Kavut, and M. D. Yücel. Balanced Boolean function on 13-variables having nonlinearity greater than the bent concatenation bound. In O. Masnyk-Hansen, J-F. Michon, P. Valarcher, and J-B. Yunès, editors, *Boolean Functions: Cryptography and Applications – BFCA 2008*, Copenhagen, Denmark, May 2008.

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

[38] S. Maitra and P. Sarkar. Cryptographically significant Boolean functions with five valued Walsh spectra. *Theoretical Computer Science*, 276(1–2):133–146, 2002.

[39] S. Maitra and P. Sarkar. Modifications of Patterson-Wiedemann functions for cryptographic applications. *IEEE Transactions on Information Theory*, 48(1):278–284, January 2002.

[40] S. Maitra, S. Sarkar, and D. K. Dalai. On dihedral group invariant Boolean functions. In J-F. Michon, P. Valarcher, and J-B. Yunès, editors, *Boolean Functions: Cryptography and Applications – BFCA 2007*, Paris, France, 2007.

[41] M. Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT 1993*, volume 765 of *LNCS*, pages 386–397. Springer-Verlag, 1994.

[42] A. Maximov. Classes of plateaued rotation symmetric Boolean functions under transformation of Walsh spectra. In *Workshop on Coding and Cryptography, WCC 2005*. Available at IACR eprint server, http://eprint.iacr.org/2004/354.ps, Dec 2004, last accessed date: August 2008.

[43] A. Maximov, M. Hell, and S. Maitra. Plateaued rotation symmetric Boolean functions on odd number of variables. In *First Workshop on Boolean Functions: Cryptography and Applications, BFCA 05*, pages 121–134, LIFAR, University of Rouen, France, March 2005.

[44] W. Millan, A. Clark, and E. Dawson. An effective genetic algorithm for finding highly nonlinear Boolean functions. In *First International Conference on Information and Communications Security*, volume 1334 of *LNCS*, pages 149–158, Springer-Verlag, 1997.

[45] W. Millan, A. Clark, and E. Dawson. Heuristic design of cryptographically strong balanced Boolean functions. In *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *LNCS*, pages 489–499, Springer-Verlag, 1998.

[46] W. Millan, A. Clark, and E. Dawson. Boolean function design using hill climbing methods. In *4th Australasian Conference on Information, Security and Privacy*, volume 1587 of *LNCS*, pages 1–11, Springer-Verlag, April 1999.

[47] J. J. Mykkeltveit. The covering radius of the $(128, 8)$ Reed-Muller code is 56. *IEEE Transactions on Information Theory*, 26(3):359–362, May 1980.

[48] National Institute of Standards and Technology. Data Encryption Standard (DES). *Federal information processing standards publication 46-2*, December 1993. Available at http://www.itl.nist.gov/fipspubs/fip46-2.htm, last accessed date: August 2008.

[49] National Institute of Standards and Technology. Advanced Encryption Standard (AES). *Federal information processing standards publication 197*, November 2001. Available at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, last accessed date: August 2008.

[50] E. Pasalic and T. Johansson. Further results on the relation between nonlinearity and resiliency of Boolean functions. In *IMA Conference on Cryptography and Coding*, volume 1746 of *LNCS*, pages 35–45, Springer-Verlag, 1999.

[51] E. Pasalic, S. Maitra, T. Johansson, and P. Sarkar. New constructions of resilient and correlation immune Boolean functions achieving upper bound on nonlinearity. In *Workshop on Coding and Cryptography - WCC 2001*, volume 6 of *Electronic Notes in Discrete Mathematics*, pages 481–490, Elsevier, Paris, France, January 2001.

[52] N. J. Patterson and D. H. Wiedemann. The covering radius of the $(2^{15}, 16)$ Reed-Muller code is at least 16276. *IEEE Transactions on Information Theory*, 29(3):354–356, May 1983.

[53] J. Pieprzyk and C. X. Qu. Fast hashing and rotation-symmetric functions. *Journal of Universal Computer Science*, 5(1):20–31, 1999.

[54] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle. Propagation characteristics of Boolean functions. In *Advances in Cryptology – EUROCRYPT 1990*, volume 473 of *LNCS*, pages 161–173, Springer-Verlag, 1991.

[55] F. S. Roberts. *Applied Combinatorics*. Prentice-Hall, Inc., Englewood Ciffs, New Jersey, 1984.

[56] O. S. Rothaus. On bent functions. *Journal of Combinatorial Theory, Series A*, 20:300–305, 1976.

[57] R. A. Rueppel. *Analysis and Design of Stream Ciphers*. Springer-Verlag, 1986.

[58] Z. Saber, M. F. Uddin, and A. Youssef. On the existence of $(9, 3, 5, 240)$ resilient functions. *IEEE Transactions on Information Theory*, 52(5):2269–2270, May 2006.

[59] P. Sarkar and S. Maitra. Construction of nonlinear Boolean functions with important cryptographic properties. In *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 485–506. Springer-Verlag, 2000.

[60] P. Sarkar and S. Maitra. Nonlinearity bounds and constuction of resilient Boolean functions. In *Advances in Cryptology – Crypto 2000*, volume 1880 of *LNCS*, pages 515–532. Springer-Verlag, 2000.

[61] S. Sarkar and S. Maitra. Idempotents in the neighbourhood of Patterson-Wiedemann functions having Walsh spectra zeros. *Designs, Codes and Cryptography*, available at http://dx.doi.org/10.1007/s10623-008-9181-y, 2008, last accessed date: August 2008.

[62] J. Seberry, X. M. Zhang, and Y. Zheng. Nonlinearly balanced Boolean functions and their propagation characteristics. In *Advances in Cryptology – CRYPTO'93*, volume 773 of *LNCS*, pages 49–60. Springer-Verlag, 1994.

[63] C. E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28:656–715, 1949.

[64] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, 30(5):776–780, September 1984.

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

[66] P. Stanica and S. Maitra. A constructive count of rotation symmetric functions. *Information Processing Letters*, 88:299–304, 2003.

[67] P. Stanica and S. Maitra. Rotation symmetric Boolean functions – count and cryptographic properties. In *R. C. Bose Centenary Symposium on Discrete Mathematics and Applications*, volume 15 of *Electronic Notes in Discrete Mathematics*, pages 178–183, Elsevier, December 2002.

[68] P. Stanica, S. Maitra, and J. Clark. Results on rotation symmetric bent and correlation immune Boolean functions. In *Fast Software Encryption Workshop – FSE 2004*, volume 3017 of *LNCS*, pages 161–177, Springer-Verlag, New Delhi, INDIA, 2004.

[69] G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Journal American Institute of Electrical Engineers*, 55(6):109–115, 1926.

[70] A. F. Webster and S. E. Tavares. On the design of S-boxes. In *Advances in Cryptology – Crypto 1985*, volume 218 of *LNCS*, pages 523–534. Springer-Verlag, 1986.

[71] M. D. Yücel. Alternative nonlinearity criteria for Boolean functions, Electrical and Electronics Engineering Department, Middle East Technical University, Ankara, Turkey, Memorandum No. 2001-1, January 2001. Available at http://www.eee.metu.edu.tr/ yucel/AlternativeNonlinearityCriteria.PDF, last accessed date: August 2008.

[72] X. M. Zhang and Y. Zheng. GAC – the criterion for global avalanche characteristics of cryptographic functions. *Journal of Universal Computer Science*, 1(5):316–333, 1995.

APPENDIX A

TRUTH TABLES OF 11- AND 13-VARIABLE DSBFs

We present the truth tables in hexadecimal format. The following is the truth table of $(11, -1, 9, 994, 200)$ DSBF.

```
68C1F052AA14260999DD0365487844C6C397A7B6114A787724957BC46471F12D
F05F873ECC6E8A29034265887BD17A2A483583367B8FF1312C347E12FA1708F3
AA1433AF952B5BE9B5F02CA891985C92114A640C2D6380C57B9BB3027E991D8D
34C45B66D00E5B7D6ACF80EEAB021A430CE54E707AAD520DAB9D472F4081FF1F
89CC06215F1B8CAA973658CF27DAADD3CF36AA0118B0DDC08716D3D526E4C70D
065371D97C2054E458A2390BD550E5736ADAC2DF8B0A10492BACC3C317B381F7
1A21F52076CB3C3DB60144F836DF2AB32DDDE0EAC051FCBD8C8F10491299751F
41F0E96761AC6F053F888DE7234945F79C9B92B3703B19BF6545C557BBBF57FF
```

The following is the truth table of $(13, -1, 10, 4036, 208)$ DSBF, which is invariant under the permutation

$$\pi(x_0, x_1, \ldots, x_{12}) = (x_0, x_2, x_4, x_6, x_8, x_{10}, x_{12}, x_1, x_3, x_5, x_7, x_9, x_{11}). \tag{A.1}$$

```
177E7EF97EFCFF937FF8EBA0FAFBC71A7EFBEAD0EC8B8815EA99FADEA12A568D
7EE8EA8BF889B215FDB1848F80950677EDC883D3AE9DB2ED9D031888277CD4F7
7FEDE881ECDC948AFF90D0968B0C0676EFE3CE028524D4FAC114C666116C2A6B
F8E2A195815AF71A89FCD3A29B48BDE3C7F6155F139090904C2B2AA1F321AA3F
7EEEF8B2E881D107ECA1E3B5C665D088EAAE9354A710C37C81CB04E4156C3A28
ECAFEC0AB5ED504C85361D75B325AA88F4560730A4386C7C13537CF04CCD299B
FA85B81D9C129772D143368CFE2A43C88096AEB4B35E8809D3DE64959BE7A90E
A12BBF7D077227FA034FC601D340931535A159CF4C88CC17BB0B4D13C8990BAE
7EFCE8ACEAC18B0DE881C517E253407FF8F4C917EC5E9F32A12C3826F700C081
F889C9F8934B2770DC7B5710F44F2EF09146A5CA1530BD3107663CA14BCD0C81
F9A18DABB9F411C88A26ECF6364474B484321F7D47E33B779B5E58679CCD85D5
FA35626C042E4E419C244A902CF12EF5420E660B6EA0EE0570A0A4B64D86979E
FFCDD1728BC516A786F10348976A7F09B212350A0A78D5F1BFA85CD8350BA194
C015D72899F98F208E1B73E9C0950093B24AE7B96C65933782CAFC7BCCD715BC
9D0208CA8AAA7FE2147A2E49192BFBCD145A74FAF0790003E75B7451930B1736
1E76880372D3A1AB70F18590E1F5177A8E8B449F61B2075AF08597D6519ECCE9
7EEDEEF5FC90CDB4ED8CA10785CE10E3E881C147F523072FB81D274B71403EEF
FB81AE60F4C6122EB9A032EC96AA5A09C8171CF41ED05879BE7F5444A101D107
```

```
EAC481D3A197ABC0825E349E192A7A40A3A07BCA367F5300EE3424AE5DECAB15
C347647CD962E09806265E01DEA75B12117F3C3C1BA1D85771DBA0A751B58512
EE82891381B78D8FCE97AE741242F081C19C593CEDF4EB2C5A3874753A619B21
D1315E5802AE6AE6247FB95F5ECB6B3FC78A22A962842D2F82A1A0A3C026B276
FBDC1B632D1878F144700CFD70EC711687F05D3025D9870548B5AB5708EDAF76
700800F86C2951DB6CECCD01BDED51766E01CD11CD349F7C21A7943CC37ED6A9
EBAFE4B6B2133A49819BF1334739D86BD128EA42504A60C1876E6CCD6FEA11D3
8A1D17180E6650C810D86FD0A622FA179BAED88422E1E3D45F6651CFDD03D734
A0151733A72E48C196D6BE92C4EF4951D5EC028A2F5FE997B00182330101824E
DE5934CDAC2FCAD63CF43D33871F0B3EC00CA1DDAAB17BDEA1B5B67E13669EE1
93A6454915D5B5C9D088C8893AABB85D07742AC84CBC20C752C2099EFADBF1F6
077532896E64AB9DFA003F974105110AED6B238E6E753716821F05DE176E5A69
52B97F79C081414F3E08A60BC816CDDE3F41AA17C0779350B912AF76073E7AC9
C5FD819B602186BE79078F5C543E36C9BF158126D33EE6697712D6A9F4E1E997
```

APPENDIX B

TRUTH TABLE OF 13-VARIABLE BALANCED FUNCTION

The following is the truth table of $(13, 0, 11, 4036, 536)$ Boolean function.

```
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E050816DF2126F6766
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E050816DF2126F6766
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E050816DF2126F6766
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E050816DF2126F6766
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E050816DF2126F6766
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E050816DF2126F6766
8B32434A935FE9AFC9EA6018E28874B7BC6BF1AA8D9F10374CEC58308B6BC267
972B23EC5D4BB6DBFA038D3E65392327024955F018D3871FAF7E920DED909899
8B32434A935FE9AFC9EA6018E28874B7BC6BF1AA8D9F10374CEC58308B6BC267
972B23EC5D4BB6DBFA038D3E65392327024955F018D3871FAF7E920DED909899
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E050816DF2126F6766
8B32434A935FE9BFC9EA6018E28874B7BC6BF3AA8D9F10374DEC59308B6BC267
972B23EC5D4BF6DBFA038D3E65392327024955F018D3A71FAF7E920DED909899
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E050816DF2126F6766
8B32434A935FE9AFC9EA6018E28874B7BC6BF1AA8D9F10374CEC58308B6BC267
972B23EC5D4BB6DBFA038D3E65392327024955F018D3871FAF7E920DED909899
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E050816DF2126F6766
8B32434A935FE9AFC9EA6018E28874B7BC6BF1AA8D9F10374CEC58308B6BC267
972B23EC5D4BB6DBFA038D3E65392327024955F018D3871FAF7E920DED909899
8B32434A935FE9AFC9EA6018E28874B7BC6BF1AA8D9F10374CEC58308B6BC267
972B23EC5D4BB6DBFA038D3E65392327024955F018D3871FAF7E920DED909899
74CDBCB56CA0165036159FE71D778B4843940E557260EFC8B313A7CF74943D98
68D4DC13A2B4492405FC72C19AC6DCD8FDB6AA0FE72C78E054816DF2126F6776
```

APPENDIX C

ALGORITHM USED IN THE CLASS OF 9-VARIABLE RSBFs

The following is the C programming code (compiled with Microsoft Visual C++ 6.0) of our steepest-descent-like iterative search algorithm that is used to find 9-variable RSBFs having nonlinearity of 241.

```
#include "stdlib.h"
#include "stdio.h"
int gn=60,R[60][9],p[]={1, 2, 3, 4, 5, 6, 7, 8, 0};
int n=9,B[512][9],h,G[9][9],A[60][60],A2[60][60][2];
long double FW2[513];

int main(){
  void orbit(int k);
  void repres();
  void walsh(int *TT, int *FW);
  void tohex(int *TT, int *tt);
  void walsheff(int *FWnew, int *FWold, int k1, int k0);
  int findmaxwh(int *FW);
  long double sumsse(int *FW);
  int i,j,k,l,t,NL,N=60000,indx,K,CHK,cnt,*P3,*Q3,srt,k0,k1;
  int FW[512],FWupd[512],tt[2];
  long double I,Maxi,cost,Q2[60];
  P3=(int *)malloc(120000*sizeof(int));
  Q3=P3;
  for (i=0;i<gn;i++)
    for (j=0;j<gn;j++)
      A[i][j]=0;
  repres();
  for (i=0;i<gn;i++){
    orbit(i);
    for (j=0;j<gn;j++)
      for (k=0;k<h;k++){
        l=0;
        for (t=0;t<n;t++)
          l=l^(G[k][t]*R[j][t]);
        A[i][j]+=1-2*l;
      }
  }
  for (i=0;i<gn;i++)
    for (j=0;j<gn;j++){
      A2[i][j][0]=2*A[i][j];
      A2[i][j][1]=-2*A[i][j];
    }
  I=0;
  for (i=0;i<513;i++){
    FW2[i]=(I*I-512)*(I*I-512);
    I+=1;
  }
  //The algorithm starts with the following truth table (randomly
  //chosen rotation symmetric truth table).
  int TT[]={0,0,1,0,1,1,1,1,1,0,0,0,1,1,0,1,0,0,1,0,0,0,1,1,1,1,0,0,
  1,1,0,1,0,1,0,1,1,0,1,1,1,0,0,1,1,1,0,1,1,0,1,0,0,0,1,1,1,0,0,0};
  walsh(TT,FW);
  tohex(TT,tt);
  NL=256-findmaxwh(FW)/2;
  printf("\nStarting... nonlinearity=%d cost=%f",NL,sumsse(FW));
  cnt=0;
  srt=gn/32+1;
  for (i=0;i<srt;i++){
    *(Q3+cnt)=tt[cnt];
    cnt+=1;
  }
  for (K=1;K<=N;K++){
    Maxi=1.6e+50;
    for (i=1;i<gn;i++){
      k0=i;
      k1=TT[i]^1;
      walsheff(FWupd,FW,k1,k0);
      cost=sumsse(FWupd);
      if (cost<=Maxi){
        Maxi=cost;
        indx=i;
      }
      Q2[i]=cost;
    }
    TT[indx]=1^TT[indx];
    tohex(TT,tt);
    for (i=0;i<K;i++){
      CHK=0;
      for (j=0;j<srt;j++)
        if (*(Q3+i*srt+j)==tt[j])
          CHK+=1;
        else
          break;
      if (CHK==srt)
        break;
    }
    while (CHK==srt){
      Maxi=1.69e+50;
      TT[indx]=1^TT[indx];
      Q2[indx]=1.7e+50;
      for (i=1;i<gn;i++)
        if (Q2[i]<=Maxi){
          Maxi=Q2[i];
          indx=i;
        }
      TT[indx]=1^TT[indx];
      tohex(TT,tt);
      for (i=0;i<K;i++){
        CHK=0;
        for (j=0;j<srt;j++)
          if (*(Q3+i*srt+j)==tt[j])
            CHK+=1;
          else
            break;
        if (CHK==srt)
          break;
      }
    }
    for (i=0;i<srt;i++){
      *(Q3+cnt)=tt[i];
      cnt+=1;
    }
    walsh(TT,FW);
    NL=256-findmaxwh(FW)/2;
    if (NL>240){
      printf("\nIteration Step=%d Nonlinearity=%d cost=%f\n",K,NL,
      sumsse(FW));
      for (i=0;i<gn;i++)
        printf("%d",TT[i]);
    }
  }
  return 0;
}

void walsh(int *TT, int *FW){
  int i,j;
  for (i=0;i<gn;i++){
    FW[i]=0;
    for (j=0;j<gn;j++)
      FW[i]=FW[i]+(1-2*TT[j])*A[j][i];
  }
}

void walsheff(int *FWnew, int *FWold, int k1, int k0){
  int i;
  for (i=0;i<gn;i++)
```

```
    FWnew[i]=FWold[i];
  for (i=0;i<gn;i++)
    FWnew[i]=FWnew[i]+A2[k0][i][k1];
}

void tohex(int *TT, int *tt){
  int i,j;
  for (i=0;i<(gn/32);i++){
    tt[i]=0;
    for (j=31;j>=0;j--)
      tt[i]=(TT[i*32+j]<<(31-j))^tt[i];
  }
  tt[gn/32]=0;
  for (j=gn-(gn/32)*32-1;j>=0;j--)
    tt[gn/32]=(TT[(gn/32)*32+j]<<(gn-(gn/32)*32-1-j))^tt[gn/32];
}

int findmaxwh(int *FW){
  int i,D,Maxi=-1;
  for (i=0;i<gn;i++){
    D=FW[i];
    if (FW[i]<0)
      D=-FW[i];
    if (D>Maxi)
      Maxi=D;
  }
  return Maxi;
}

long double sumsse(int *FW){
  int i;
  long double sum=0;
  for (i=0;i<gn;i++){
    if (FW[i]<0)
      sum=sum+FW2[-FW[i]];
    else
      sum=sum+FW2[FW[i]];
  }
  return sum;
}

void orbit(int k){
  int i,j,l,chk=1,b[9],a[9];
  h=0;
  for (i=0;i<n;i++){
    G[h][i]=R[k][i];
    b[i]=R[k][i];
  }
  while (chk!=0){
    h+=1;
    for (i=0;i<n;i++)
      a[i]=b[p[i]];
    for (i=0;i<n;i++)
      b[i]=a[i];
    for (i=0;i<h;i++){
      l=0;
      for (j=0;j<n;j++)
        if (G[i][j]==a[j])
          l+=1;
      if (l==n){
        chk=0;
        break;
      }
    }
    if (chk!=0)
      for (i=0;i<n;i++)
        G[h][i]=a[i];
  }
}

void repres(){
  int i,j,k=0;
  for (i=0;i<512;i++){
    for (j=0;j<n;j++)
      B[i][n-1-j]=(i&(1<<j))>>j;
  }
  for (i=0;i<512;i++)
    if (B[i][0]!=-1){
      for (j=0;j<n;j++)
        R[k][j]=B[i][j];
      orbit(k);
      for (j=0;j<h;j++)
        B[256*G[j][0] + 128*G[j][1] + 64*G[j][2] + 32*G[j][3] +
        16*G[j][4] + 8*G[j][5] + 4*G[j][6] + 2*G[j][7] +
        1*G[j][8]][0]=-1;
      k+=1;
    }
}
```

APPENDIX D

ALGORITHM USED IN THE CLASS OF 9-VARIABLE DSBFs

The following is the C programming code (compiled with Microsoft Visual C++ 6.0) of our steepest-descent-like iterative search algorithm that is used to find 9-variable DSBFs having nonlinearity of 242.

```
#include "stdlib.h"
#include "stdio.h"
int R[104][9],gn=104,p[]={3, 4, 5, 6, 7, 8, 0, 1, 2};
int n=9,B[512][9],h,G[6][9],A[104][104],A2[104][104][2];
long double FW2[513];

int main(){
  void orbitd(int k);
  void repres();
  void walsh(int *TT, int *FW);
  void tohex(int *TT, int *tt);
  void walsheff(int *FWnew, int *FWold, int k1, int k0);
  int findmaxwh(int *FW);
  long double sumsse(int *FW);
  int i,j,k,l,t,NL,N=60000,indx,K,CHK,cnt,*P3,*Q3,srt,k0,k1;
  int FW[512],FWupd[512],tt[4];
  long double I,Maxi,cost,Q2[104];
  P3=(int *)malloc(240000*sizeof(int));
  Q3=P3;
  for (i=0;i<gn;i++)
    for (j=0;j<gn;j++)
      A[i][j]=0;
  repres();
  for (i=0;i<gn;i++){
    orbitd(i);
    for (j=0;j<gn;j++)
      for (k=0;k<h;k++){
        l=0;
        for (t=0;t<n;t++)
          l=l^(G[k][t]*R[j][t]);
        A[i][j]+=1-2*l;
      }
  }
  for (i=0;i<gn;i++)
```

```
 for (j=0;j<gn;j++){
  A2[i][j][0]=2*A[i][j];
  A2[i][j][1]=-2*A[i][j];
 }
I=0;
for (i=0;i<513;i++){
 FW2[i]=(I*I-512)*(I*I-512);
 I+=1;
}
//The algorithm starts with the following truth table (randomly
//chosen 3-dihedral symmetric truth table).
int TT[]={0,0,1,0,1,1,0,1,0,0,0,0,0,0,1,1,1,1,1,0,1,0,1,1,0,1,1,0,
0,0,1,0,1,1,0,1,0,0,0,0,1,0,1,0,0,0,0,1,1,0,0,1,0,0,0,1,1,1,0,1,0,
1,1,0,1,1,0,0,0,0,1,0,1,1,1,1,0,0,0,0,1,1,0,1,0,0,0,0,1,1,1,0,1,1,
1,0,0,1,1,1,0,0,0,0};
walsh(TT,FW);
tohex(TT,tt);
NL=256-findmaxwh(FW)/2;
printf("\nStarting... nonlinearity=%d cost=%f",NL,sumsse(FW));
cnt=0;
srt=gn/32+1;
for (i=0;i<srt;i++){
 *(Q3+cnt)=tt[cnt];
 cnt+=1;
}
for (K=1;K<=N;K++){
 Maxi=1.6e+50;
 for (i=1;i<gn;i++){
  k0=i;
  k1=TT[i]^1;
  walsheff(FWupd,FW,k1,k0);
  cost=sumsse(FWupd);
  if (cost<=Maxi){
   Maxi=cost;
   indx=i;
  }
  Q2[i]=cost;
 }
 TT[indx]=1^TT[indx];
 tohex(TT,tt);
 for (i=0;i<K;i++){
  CHK=0;
  for (j=0;j<srt;j++)
   if (*(Q3+i*srt+j)==tt[j])
    CHK+=1;
   else
    break;
  if (CHK==srt)
   break;
```

```
    }
    while (CHK==srt){
      Maxi=1.69e+50;
      TT[indx]=1^TT[indx];
      Q2[indx]=1.7e+50;
      for (i=1;i<gn;i++)
        if (Q2[i]<=Maxi){
          Maxi=Q2[i];
          indx=i;
        }
      TT[indx]=1^TT[indx];
      tohex(TT,tt);
      for (i=0;i<K;i++){
        CHK=0;
        for (j=0;j<srt;j++)
          if (*(Q3+i*srt+j)==tt[j])
            CHK+=1;
          else
            break;
        if (CHK==srt)
          break;
      }
    }
    for (i=0;i<srt;i++){
      *(Q3+cnt)=tt[i];
      cnt+=1;
    }
    walsh(TT,FW);
    NL=256-findmaxwh(FW)/2;
    if (NL>241){
      printf("\nIteration step=%d nonlinearity=%d cost=%f\n",K,NL,
      sumsse(FW));
      for (i=0;i<gn;i++)
        printf("%d",TT[i]);
    }
  }
  return 0;
}

void walsh(int *TT, int *FW){
  int i,j;
  for (i=0;i<gn;i++){
    FW[i]=0;
    for (j=0;j<gn;j++)
      FW[i]=FW[i]+(1-2*TT[j])*A[j][i];
  }
}

void walsheff(int *FWnew, int *FWold, int k1, int k0){
  int i;
  for (i=0;i<gn;i++)
    FWnew[i]=FWold[i];
  for (i=0;i<gn;i++)
    FWnew[i]=FWnew[i]+A2[k0][i][k1];
}

void tohex(int *TT, int *tt){
  int i,j;
  for (i=0;i<(gn/32);i++){
    tt[i]=0;
    for (j=31;j>=0;j--)
      tt[i]=(TT[i*32+j]<<(31-j))^tt[i];
  }
  tt[gn/32]=0;
  for (j=gn-(gn/32)*32-1;j>=0;j--)
    tt[gn/32]=(TT[(gn/32)*32+j]<<(gn-(gn/32)*32-1-j))^tt[gn/32];
}

int findmaxwh(int *FW){
  int i,D,Maxi=-1;
  for (i=0;i<gn;i++){
    D=FW[i];
    if (FW[i]<0)
      D=-FW[i];
    if (D>Maxi)
      Maxi=D;
  }
  return Maxi;
}

long double sumsse(int *FW){
  int i;
  long double sum=0;
  for (i=0;i<gn;i++){
    if (FW[i]<0)
      sum=sum+FW2[-FW[i]];
    else
      sum=sum+FW2[FW[i]];
  }
  return sum;
}

void orbitd(int k){
  int i,j,l,chk=1,b[9],a[9],h2;
  h=0;
  for (i=0;i<n;i++){
    G[h][i]=R[k][i];
    b[i]=R[k][i];
```

```
}

while (chk!=0){
  h+=1;
  for (i=0;i<n;i++)
    a[i]=b[p[i]];
  for (i=0;i<n;i++)
    b[i]=a[i];
  for (i=0;i<h;i++){
    l=0;
    for (j=0;j<n;j++)
      if (G[i][j]==a[j])
        l+=1;
    if (l==n){
      chk=0;
      break;
    }
  }
  if (chk!=0)
    for (i=0;i<n;i++)
      G[h][i]=a[i];
}

chk=1;
for (i=0;i<n;i++)
  b[i]=G[0][n-i-1];
for (i=0;i<h;i++){
  l=0;
  for (j=0;j<n;j++)
    if (G[i][j]==b[j])
      l+=1;
  if (l==n){
    chk=0;
    break;
  }
}
if (chk!=0)
  for (i=0;i<n;i++)
    G[h][i]=b[i];
h2=1;
while (chk!=0){
  h+=1;
  for (i=0;i<n;i++)
    a[i]=G[h2][n-i-1];
  h2+=1;
  for (i=0;i<n;i++)
    b[i]=a[i];
  for (i=0;i<h;i++){
    l=0;
```

```
      for (j=0;j<n;j++)
        if (G[i][j]==b[j])
          l+=1;
      if (l==n){
        chk=0;
        break;
      }
    }
    if (chk!=0)
      for (i=0;i<n;i++)
        G[h][i]=a[i];
  }

}

void repres(){
  int i,j,k=0;
  for (i=0;i<512;i++){
    for (j=0;j<n;j++)
      B[i][n-1-j]=(i&(1<<j))>>j;
  }
  for (i=0;i<512;i++)
    if (B[i][0]!=-1){
      for (j=0;j<n;j++)
        R[k][j]=B[i][j];
      orbitd(k);
      for (j=0;j<h;j++)
        B[256*G[j][0] + 128*G[j][1] + 64*G[j][2] + 32*G[j][3] +
        16*G[j][4] + 8*G[j][5] + 4*G[j][6] + 2*G[j][7] +
        1*G[j][8]][0]=-1;
      k+=1;
    }
}
```

VITA

PERSONAL INFORMATION

Surname, Name: Kavut, Selçuk
Nationality: Turkish (TC)
Date and Place of Birth: 28 April 1978 , Ankara
Marital Status: Married
Phone: +90 312 210 45 83
Fax: +90 312 210 23 04
email: kavut@eee.metu.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	Electrical and Electronics Engineering, METU	2002
BS	Electronics Engineering, Ankara University	1998

WORK EXPERIENCE

Year	Place	Enrollment
1999-Present	Electrical and Electronics Engineering, METU	Research Assistant
1998-1999	SDG Yapı Endüstri Tesisleri Sanayi ve Ticaret A.Ş.	Electronics Engineer

PUBLICATIONS

1. S. Kavut and M. D. Yücel, "Random Permutations on Input Vectors of Boolean Functions", Boolean Functions: Cryptography and Applications – BFCA 2008, Editors: O. Masnyk-Hansen, J-F. Michon, P. Valarcher, J-B. Yunès, Copenhagen, Denmark, May 2008.

2. S. Maitra, S. Kavut and M. D. Yücel, "Balanced Boolean Function on 13-Variables having Nonlinearity greater than the Bent Concatenation Bound", Boolean Functions: Cryptography and Applications – BFCA 2008, Editors: O. Masnyk-Hansen, J-F. Michon, P. Valarcher, J-B. Yunès, Copenhagen, Denmark, May 2008.

3. S. Kavut and M. D. Yücel. "Generalized Rotation Symmetric and Dihedral Symmetric Boolean Functions – 9 variable Boolean Functions with Nonlinearity 242", 17th International Applied Algebra, Algebraic Algorithms, and Error Correcting Codes (AAECC-17) Symposium, Editors: Serdar Boztaş, Hsiao-Feng (Francis) Lu, LNCS Vol. 4851, pages 321-329, Bangalore, India, December 2007.

4. S. Kavut, S. Maitra and M. D. Yücel. "Search for Boolean Functions with Excellent Profiles in the Rotation Symmetric Class", IEEE Transactions on Information Theory, Volume: 53, Issue: 5, pages 1743-1751, ISSN: 0018-9448, May 2007.

5. S. Kavut, M. D. Yücel and S. Maitra. "Construction of Resilient Functions by the Concatenation of Boolean Functions Having Nonintersecting Walsh Spectra", Boolean Func-

tions: Cryptography and Applications – BFCA 2007, Editors: J-F. Michon, P. Valarcher, J-B. Yunès, Paris, France, May 2007.

6. S. Kavut, S. Maitra, S. Sarkar and M. D. Yücel. “Enumeration of 9-variable Rotation Symmetric Boolean Functions having Nonlinearity > 240”, Progress in Cryptology – Indocrypt 2006, Editors: R. Barua and T. Lange, LNCS Vol. 4329, Springer-Verlag, pages 266-279, Kolkata, India, December 2006.

7. S. Kavut and M. D. Yücel. “Finite Field Power Mappings in Cryptography”, National Cryptology Symposium II, pages 37-49, Ankara, Turkey, December 2006.

8. S. Kavut, S. Maitra, S. Sarkar and M. D. Yücel. “Autocorrelation Spectra of Balanced Boolean Functions on An Odd Number of Input Variables with Maximum Absolute Value $< 2^{\frac{n+1}{2}}$”, Boolean Functions: Cryptography and Applications – BFCA 2006, Editors: J-F. Michon, P. Valarcher, J-B. Yunès, pages 73-86, Rouen, France, March 2006.

9. S. Kavut and M. D. Yücel. “A New Algorithm in the Design of Strong Boolean Functions” (in Turkish), National Cryptology Symposium I, pages 95-105, Ankara, Turkey, November 2005.

10. S. Kavut and M. D. Yücel. “Improved Cost Function in the Design of Boolean Functions Satisfying Multiple Criteria”, Progress in Cryptology – Indocrypt 2003, Editors: S. Maitra and T. Johansson, LNCS Vol. 2904, Springer-Verlag, pages 121-134, New Delhi, India, December 2003.

11. S. Kavut and M. D. Yücel. “Slide Attack on Spectr-H64”, Progress in Cryptology – Indocrypt 2002, Editors: A. Menezes and P. Sarkar, LNCS Vol. 2551, Springer-Verlag, pages 34-47, Hyderabad, India, December 2002.

12. S. Kavut and M. D. Yücel. “Rijndael Algoritmasının Kriptografik Özellikleri”, İletişim Teknolojileri 1. Ulusal Sempozyum ve Fuarı, pages 289-294, ODTÜ Kültür Kongre Merkezi, Ankara, Turkey, 17-21 October, 2001.

13. S. Kavut and M. D. Yücel. “On Some Cryptographic Properties of Rijndael”, Information Assurance in Computer Networks, Methods, Models and Architectures for Network Security, Editors: V. I. Gorodetski, V. A. Skormin, L. J. Popyack, LNCS Vol. 2052, Springer-Verlag, pages 300-311, St. Petersburg, Russia, May 2001.