On Some Necessary Conditions of Boolean Functions to Resist Algebraic Attacks

Deepak Kumar Dalai

Applied Statistics Unit
Indian Statistical Institute
Kolkata, India
August, 2006

On Some Necessary Conditions of Boolean Functions to Resist Algebraic Attacks

Thesis submitted to Indian Statistical Institute in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy

by

Deepak Kumar Dalai
Applied Statistics Unit
Indian Statistical Institute
203, B. T. Road, Kolkata 700 108, INDIA
e-mail : deepak_r@isical.ac.in

under the supervision of

Dr. Subhamoy Maitra
Applied Statistics Unit
Indian Statistical Institute
203, B. T. Road, Kolkata 700 108, INDIA
e-mail : subho@isical.ac.in

To my parents.

Acknowledgments

There are many people who helped in different ways to make this thesis possible. I would like to thank them for their suggestions and advice throughout this work. Without their presence I would still be wandering.

I am extremely lucky to have worked with my supervisor Dr. Subhamoy Maitra (Subho-da) for his heartily guidance, patience and tolerance during my PhD course. I am indebted to him for all kinds of support throughout my PhD course period. It is a great pleasure for me to thank Prof. Bimal Roy (Bimal-da) who supported me in many ways (from serious study to play soccer) for my research and created opportunities to visit different places of the world and to interact with eminent scientists. I feel extremely privileged to work with the research community guided by him. I would like to thank Prof. Rana Barua (Rana-da) and Dr. Palash Sarkar (Palash-da) for their great teaching on Theoretical Computer Science and Cryptography that helped me a lot to get into cryptography research. I express my gratitude to Prof. Swadhinananda Pattanayak of Institute of Mathematics and Application, Bhubaneswar for motivating me towards research and influential teaching on mathematics during my post graduate courses in Mathematics. To each of my teachers I owe a great debt of gratitude for their wonderful teaching which worked as stairs to reach at this stage.

Thanks to Prof. Claude Carlet for his interesting discussions at INRIA, Paris and continuous email communications on research with me. I like to thank Dr. Kishan Chand Gupta (Kishan-da) of University of Waterloo for his continuous encouragements, academic collaboration and making funs as well. I also thank Dr. Sugata Gangopadhyay of IIT, Roorkee for his useful discussions with me.

I acknowledge Avishek, Prem and Sumanta for spending their valuable time for careful reading of an early draft of this thesis. I am very pleased to thank Avishek and Prem for their heartily friendship with whom I had a memorable time over the last five years. I would like to warmly thank Debrup-da, Dibyendu, Ipsita, Madhu, Mridul, Raja, Ratna-di, Sanjit-da, Somitra, Sourav-da and Sumanta for having great time with them. I spent many enjoyable hours with the members of cryptography research community chatting about crazy ideas in computer laboratory and over a cup of tea at the gate of our institute. My special thank to each members of this community for having such rich and friendly environment.

I would like to thank Indian Statistical Institute for providing me opportunity for the PhD and MTech programmes. I thank all the faculty and staff members of Applied Statistics Unit for allowing me to be with them and their cooperations in many ways.

Thanks to my family and uncles who have been extremely understanding and supportive to my studies. Last but not the least I am gladdest to thank my parents for their understanding, never ending blessings, faiths and supports.

Abstract

In this thesis we discuss certain properties of Boolean functions that are necessary for resistance against algebraic and fast algebraic attacks. A Boolean function $f(x_1, \ldots, x_n)$ on n variables may be described as a multivariate polynomial over $GF(2)$ and it is well known that its algebraic degree d should not be low if it has to be used as a primitive in a well designed cryptosystem. Recently, it has been noted that a necessary condition in resisting algebraic attack is as follows: the function f should not have a relation $fg = h$, where g, h are nonzero n-variable Boolean functions of low degrees. This condition boils down to the situation that the function f should not have relations like $fh_1 = 0$ or $(1 + f)h_2 = 0$, where h_1, h_2 are nonzero n-variable Boolean functions of low degrees. The function h_1 (respectively h_2) is called the annihilator of f (respectively $1 + f$). The notation $\mathsf{AI}_n(f)$ is used to denote the minimum degree of the annihilators of f or $1 + f$. This is well known as "Algebraic Immunity" of the function f in literature. There are evidences that algebraic immunity is not a sufficient condition to resist against all kinds of algebraic attacks, but clearly it is one of the most important necessary conditions. The term "Annihilator Immunity" may be a more appropriate notation than "Algebraic Immunity", but following the frequent use of the term "Algebraic Immunity" in currently available research materials, we use the term Algebraic Immunity in this thesis. It is known that $\mathsf{AI}_n(f) \leq \lceil \frac{n}{2} \rceil$.

Good nonlinearity is one of the most important properties of Boolean functions to be used in a cryptosystem. We present a fundamental relationship between the algebraic immunity and the nonlinearity of a Boolean function. We first relate the weight of a function with its algebraic immunity and then extend the result to show that if $\mathsf{nl}(f) < \sum_{i=0}^{d} \binom{n}{i}$, then $\mathsf{AI}_n(f) \leq d + 1$. That is, if $\mathsf{AI}_n(f) > d + 1$ then $\mathsf{nl}(f) \geq \sum_{i=0}^{d} \binom{n}{i}$. Thus while choosing a function with good algebraic immunity, the nonlinearity of the function is lower bounded. The main idea (in proving these results) is based on solutions to a set of homogeneous linear equations. Using similar approach, given a Boolean function, we have also studied the number of linearly independent annihilators at the lowest possible degree. Further we have studied some existing constructions of cryptographically significant Boolean functions in terms of their algebraic immunity.

As there was no known construction of Boolean function with maximum possible algebraic immunity, next we concentrate on that problem. So far, the attempt in designing Boolean functions with required algebraic immunity was only ad-hoc, i.e., the functions were designed keeping in mind the other cryptographic criteria, and then it has been checked whether the function can provide good algebraic immunity too. For the first time, we present a construction method to generate Boolean functions on n variables with highest possible

algebraic immunity $\lceil \frac{n}{2} \rceil$. The construction is recursive in nature, i.e., a function on higher number variables is built using proper functions on lower number of variables.

Further we tried to concentrate on the basic theory that how a function with maximum possible algebraic immunity can be constructed. We considered three n-variable functions f, f_1, f_2. The functions f_1, f_2 are such that they have no annihilator having degree less than $\lceil \frac{n}{2} \rceil$. Then one can construct a function f with the maximum possible algebraic immunity $\lceil \frac{n}{2} \rceil$ where $supp(f) \supseteq supp(f_2)$ and $supp(1+f) \supseteq supp(f_1)$. We applied this strategy to get functions with maximum possible algebraic immunity and specifically concentrated on the symmetric Boolean functions with such property. The algebraic degree and nonlinearity of these symmetric functions (and related non symmetric functions) are studied in detail.

It has been revealed that Boolean functions with maximum possible algebraic immunity may be weak against fast algebraic attacks. It may very well happen that given a Boolean function f with $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$, one can get functions g, h with $\deg(h) = \lceil \frac{n}{2} \rceil$ where $\deg(g)$ is very low. The low degree of g makes the function f vulnerable against certain kinds of fast algebraic attacks. This motivates us to analyse the functions f (with maximum possible algebraic immunity) and to check the degree of g under such a scenario. We study different functions (produced by our constructions, and also other functions) to check this property and identify when the situation is encouraging and when it is not.

Given a Boolean function f on n-variables, a set of homogeneous linear equations can be formed, by solving which one can decide whether there exist annihilators at degree d or not. We analyse how the number of homogeneous linear equations can be reduced and show that the reduction can be significant if one can find an affine transformation over $f(x)$ to get $h(x) = f(Bx + b)$ such that $|\{x \mid h(x) = 1, wt(x) \leq d\}|$ is maximized. We present an efficient heuristic towards this. Our study also shows for what kind of Boolean functions the asymptotic reduction is possible by this strategy and when the reduction is not asymptotic but constant. Our method helps in theoretically understanding the structure of the set of homogeneous linear equations used to find the annihilators.

List of Publications

This doctoral thesis is based on the following refereed conference and journal papers.

1. D. K. Dalai, K. C. Gupta and S. Maitra. Results on Algebraic Immunity for Cryptographically Significant Boolean Functions. In *Progress in Cryptology - Indocrypt 2004*, number 3348 in Lecture Notes in Computer Science, pages 92–106. Springer Verlag, 2004 [62].

2. D. K. Dalai, K. C. Gupta and S. Maitra. Cryptographically Significant Boolean functions: Construction and Analysis in terms of Algebraic Immunity. In *Fast Software Encryption, FSE 2005*, number 3557 in Lecture Notes in Computer Science, pages 98–111. Springer-Verlag 2005 [63].

3. C. Carlet, D. K. Dalai, K. C. Gupta and S. Maitra. Algebraic Immunity for Cryptographically Significant Boolean Functions: Analysis and Construction. *IEEE Transactions on Information Theory*, IT-52(7):3105 – 3121, 2006 [36]. (This is an extended and revised version of above two papers [62, 63]).

4. D. K. Dalai, S. Maitra and S. Sarkar. Basic Theory in Construction of Boolean Functions with Maximum Possible Annihilator Immunity. *Design, Codes and Cryptography* 40(1):41–58, July 2006 [66].

5. D. K. Dalai, K. C. Gupta and S. Maitra. Notion of Algebraic Immunity and Its evaluation Related to Fast Algebraic Attacks. In *Second International Workshop on Boolean Functions: Cryptography and Applications, BFCA* 06, March 13–15, 2006, LIFAR, University of Rouen, France [64].

6. D. K. Dalai and S. Maitra. Reducing the Number of Homogeneous Linear Equations in Finding Annihilators. Accepted in *International Conference on Sequences and Their Applications*, 2006 (SETA06), September 24 – 28, 2006 Beijing, China. To be published in number 4086 in Lecture Notes in Computer Science, Springer 2006 [65].

Contents

List of Tables

Chapter 1

Introduction

The literature of cryptography has a very long and curious history. A cryptographic scheme is considered to be secure if an attacker, who has access to the algorithmic principle of the scheme but has no knowledge about the key, is not able to attack the scheme. This cryptographic principle was first introduced by Kerckhoffs [103] in 1883. Then another influential paper [83] on cryptanalysis came in 1920 by William F. Friedman. By the end of world war II, the work of Shannon [159] was of great influence in the scientific study of cryptography. From the fourth quarter of twentieth century, the revolution of digital computers and network communication forced the military and commercial sectors to protect their information stored or transmitted in digital form. Apart from these sectors, common people have also started to depend on computers and network communication; naturally these applications also need proper security too. Due to these reasons, cryptography has become an important subject to explore. The most striking development of cryptography started with the key paper [71] by Diffie and Hellman. The reader may refer to the books [125, 167] for a more elaborate documentation.

Cryptology includes designing of cryptosystem and their analysis interms of security, complexities, performance, compatibility etc. The later one, known as cryptanalysis, mainly deals with finding weaknesses in the cryptosystems. The fundamental idea of cryptography is that any two (or more) communicating parties want to make sure that any eavesdropper can not read and/or change the information they are exchanging. Depending on the application areas, cryptography is divided into several parts. The most popular areas are as follows:

1. encryption and decryption for secret communication of message,
2. signature and message authentication code (MAC) to authenticate data,

3. key agreement protocol to agree on a secret key by a group of people.

This thesis falls in the area of item 1. For other two areas and further information see [125, 167]. Encryption and decryption are used for secret transmission of message from one end to another such that in the middle any unwanted person cannot understand the meaning of the message. The *plaintext* message M that the sender wants to transmit is considered as a sequence of characters from a set of fixed characters called *alphabet*. M is encrypted to produce another sequence of characters from the set alphabet and the encrypted sequence is called the *cipher* C. In practice, we use the binary digits (bits) 0, 1 as the alphabet. The encryption function $\mathcal{E}_{k_e}$ operates on M to produce C and the decryption function $\mathcal{D}_{k_d}$ operates on C to recover original plaintext M. Both the encryption function $\mathcal{E}_{k_e}$ and the decryption function $\mathcal{D}_{k_d}$ are parameterized by the keys k_e and k_d respectively, which are chosen from a very large set of possible keys called the *keyspace*. The sender encrypts the plaintext by computing $C = \mathcal{E}_{k_e}(M)$ and sends C to the receiver. The functions are properly designed so that the receiver recovers the original text by computing $\mathcal{D}_{k_d}(C) = \mathcal{D}_{k_d}(\mathcal{E}_{k_e}(M)) = M$.

The two major divisions in encryption/decryption strategies are as follows:

1. Secret key cryptosystems (also called symmetric key cryptosystems).

2. Public key cryptosystems (also called asymmetric key cryptosystems).

Our work is in the area of secret key cryptosystems. See [86, 125, 167] for detailed references on public key cryptosystems.

The conventional strategy, where the decryption key k_d is either the same as the encryption key k_e or easily derivable from k_e is called symmetric key cryptosystem. From now on we consider both k_e and k_d as the same single key k. These cryptosystems are also called secret key cryptosystems since both the sender and the receiver agree on a single key which is kept secret. There are two classes of design paradigms for the functions $\mathcal{E}_k(M)$ and $\mathcal{D}_k(C)$: block ciphers and stream ciphers.

First we briefly explain the block ciphers. A block cipher breaks up a plaintext M into successive blocks $M_1, M_2, \ldots$ of elements from alphabet. Each block is encrypted using a key k (same for all blocks) from the keyspace K. That is, the plaintext M is encrypted as $C_1 = \mathcal{E}_k(M_1), C_2 = \mathcal{E}_k(M_2), \ldots$. Then, at the receiver end, the plaintext M is recovered as $\mathcal{D}_k(C_1), \mathcal{D}_k(C_2), \ldots$. The block cipher literature is extremely rich and one may refer to [167, 125] for more details. The most well known block cipher at this time is the *Advanced*

Encryption Standard (AES), also known as Rijndael [132]. This is the successor of another well known cipher, DES [133]. AES was adopted by NIST in 2001 after a 5-year long standardisation process. One may also refer to some other popular block ciphers such as FEAL [160], IDEA [108], RC6 [143], SERPENT [9], TWOFISH [153] etc. Most of them use substitution boxes (in short it is called S-boxes) as the nonlinear part of the design. See [125] for more details on design and study of block ciphers. Mathematically, any S-box can be viewed as a multi-output Boolean function, i.e., a function $S : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$. So, an S-box can be treated as an ordered set of m many n variable Boolean functions.

Let us now concentrate on stream ciphers. Some recent popular stream ciphers are SNOW [76], SCREAM [92], TURING [145], MUGI [172], HBB [149], RABBIT [15], HELIX [81] etc. Some more recently proposed stream ciphers are available for the ECrypt stream cipher project [169]. The basic structure of this system is as follows. A secret sequence of bits (of length equal to the message length) is bitwise XOR-ed (addition modulo 2) with the message sequence and the resulting sequence (the cipher) is sent to the receiver. The receiver deciphers it by bitwise XOR-ing the cipher bits with the same secret sequence. The security of stream cipher depends on the unpredictability of the bits of this secret sequence. A sequence is unpredictable if it is random. In information theoretic sense, a random sequence provides *unconditional security* if the secret sequence, once used for encryption, is never used again. This is known as *one time pad* (see [167, page 50, first edition]). The main advantage of stream cipher is that if a fast random bit generator is available, then both enciphering and deciphering are very fast as we need only the XOR operation during encryption and decryption.

In practice, one can generate a pseudorandom sequence using some algorithm and a small secret key is used to initialize the algorithm. A part of the sequence is used to encipher the message. At the other end same secret key and the same algorithm are used to regenerate the correct part of the secret sequence to decipher. So, the problem in stream cipher design is to construct a good pseudorandom generator. One may refer to [105, 14, 88] for design of pseudorandom generators to be used in stream cipher systems.

Linear Feedback Shift Registers (LFSRs) [88] are one of the most used primitives in pseudorandom generators. The sequences generated from several LFSRs are combined by nonlinear combiners, generally nonlinear Boolean functions, to introduce the nonlinearity (see [147]). The standard model of pseudorandom generator [60, 73, 161, 162] combines the outputs of several independent LFSR sequences using a nonlinear Boolean function to produce the keystream.

Before accepting a cryptosystem for encryption, one must study the underlying algorithm

to analyse the security. Following the Kerckhoff's principle [103], designer must accept that attacker knows the algorithm of the system, i.e., during the cryptanalysis the adversary has complete knowledge of the encryption algorithm used, along with the parameters of the system; only the secret key remains unknown. Let us now give a brief introduction on four general cryptanalytic attacks.

1. Ciphertext Only Attacks : The adversary has only the ciphertext of several messages, all of which have been encrypted using same algorithm. From this the adversary has to recover the message as much as possible or, better to extract the secret key (or keys) used.

2. Known Plaintext Attack : The adversary possesses several message texts along with the corresponding ciphertexts. This is a more advantageous assumption to the cryptanalyst than the previous case.

3. Chosen Plaintext Attack : This is a variation of the previous attack where the adversary is allowed to choose the set of message texts. This is more powerful than the previous case in the sense that one that one may yield further information by cleverly choosing some selected plaintext strings. It can be either passive (prior decision is taken which strings to be selected) or adaptive (decision is taken during encryption looking at the results of the previously chosen strings).

4. Chosen Ciphertext Attack : This is an another variation of known plain text attack. The adversary can choose some ciphertexts and get temporary access to decrypt it to have corresponding plaintexts. Like the previous attack the attacker may earn more advantages by cleverly choosing some selected ciphertexts.

On the basis of these attacks both block and stream ciphers can be analysed in several ways.

Proper choice of S-boxes is an important part in block cipher design. Matsui [119] introduced linear cryptanalysis method for block ciphers and implemented that on DES. By this technique, the linear combination of the component functions of an S-box, used in a block cipher, are approximated by linear functions of the input variables. To resist linear cryptanalysis, S-boxes should have high nonlinearity. Differential cryptanalysis [10] is a chosen-plaintext attack and involves comparing the XOR of two inputs to the XOR of corresponding two outputs. A non uniform output distribution is the basis for a successful differential attack. Motivated by this, Webster and Tavares [173] introduced the concept of strict avalanche criteria (SAC). A related property called propagation characteristics (PC)

was considered by Preneel, Leekwijck, Linden, Govaerts and Vandewalle [142]. PC and SAC are two important cryptographic properties for S-boxes to resist differential cryptanalysis. For details of these attacks see [167, second edition]. One may also refer to [154, 156, 157, 158, 90] for design of Boolean functions having good PC and SAC properties.

Jakobsen and Knudsen [97] identified interpolation attack on block ciphers with S-boxes having small algebraic degree. Later Canteaut and Videau [29] provided higher order differential attack on block ciphers using S-boxes with low algebraic degree. So algebraic degree of S-boxes should be high to resist such attacks.

In case of stream ciphers, it is important to study the properties of the underlying nonlinear functions. In last two decades several classes of attacks have been proposed on stream ciphers in this direction. Thus, the properties of nonlinear functions have received lots of attention in symmetric key cryptography literature. First, the function should be balanced to satisfy the pseudorandomness of the generated sequence. Otherwise, for a large set of randomly selected input values, the proportion of 0's and 1's in the output values of the function will be away from half and the system may become vulnerable to cryptanalytic attacks. In the stream cipher model, the linear complexity [73] of the generating keystream must be large enough and stable. High algebraic degree is a necessary condition to provide high linear complexity [148, 73]. The keystream bits can be guessed by approximating the keystream generated by an affine function and this type of attack is called *best affine approximation* (BAA) attack. A function with low nonlinearity is prone to BAA attack (see [73, Chapter 3]). A Boolean function should have high nonlinearity to be used in stream ciphers. To resist divide-and-conquer attack, a Boolean function used in stream cipher, should be correlation immune of higher order [161, 162, 73]. Some more important approaches to the cryptanalysis of LFSR based stream ciphers are available in literature, e.g., fast correlation attacks [124, 42, 98, 99], backtracking attacks [87, 178, 177], time-space trade offs [12], BDD-based attacks [107] etc. To know the details of recent works on cryptanalysis on stream ciphers one may refer to the thesis by Maximov [121].

Recently, a new class of attacks named *algebraic attacks* has become very important to cryptanalyse both block and stream cipher cryptosystems. In these kind of attacks, the attacker attempts to find a large set of algebraic equations over the secret key and the output key bits. Knowing some output key bits, the attacker attempts to recover the secret bits by solving these equations. Hence, this attack falls under the known and chosen plaintext attack and algebraic in nature rather than statistical. The efficiency of the attack depends on the efficiency of the algorithm to generate the algebraic equations and to solve the generated large set of multivariate equations. If the degree of the equations are low or the equations

are of special structures, then the system of equations may be solved efficiently.

Courtois and Pieprzyk [58] have exploited the overdefined relations between input and output bits of the block ciphers for solving the initial key bits. The algebraic attack on the stream ciphers is also interesting (see [5, 56, 51, 123]). Two fundamental models of stream ciphers are nonlinear combiner and nonlinear filter generator, where a nonlinear Boolean function f takes an important role to generate the keystream. It has been found [56, 123] that if a function f or its complement $1 + f$ has low degree annihilators (see Chapter 2, Definition 10), one can construct equations of degree equal to the degree of the annihilators. Then one may attempt to solve this system of equations to recover the secret key or to reduce the key space. For this reason the designer should be careful that the underlying function f or its complement $1 + f$ should not have low degree annihilators. Hence the property, that both f and $1 + f$ have no low degree annihilators, is necessary for choosing a Boolean function in the design of a cryptosystem. This property is called the algebraic immunity and is defined as the minimum degree of the sets of all annihilators of f and $1 + f$. This property may not resist all types of algebraic attacks (for example fast algebraic attacks [51]), but clearly it is an important necessary condition. This thesis is devoted to study the Boolean functions in terms of their algebraic immunity.

Studying the properties of underlying Boolean functions is an important task for design and cryptanalysis of both stream and block cipher systems. From the recent literature, finding a Boolean function which achieves all these desired properties appears to be a hard task and there are some trade offs among these properties. Depending on the application requirement one has to decide which properties are more important.

1.1 Thesis Organization

The current chapter (Chapter 1) discusses some introductory materials. In Chapter 2 we present related background information for this thesis. There we introduce the definitions and properties of Boolean functions which are useful to our work. Then we present an overview on the literature on algebraic attacks, from which we derive the motivation towards this thesis.

Chapter 3 initiates our work in the area of algebraic immunity of Boolean functions. We relate algebraic immunity with weight and nonlinearity of Boolean functions in Section 3.1. Then we present some results on enumeration of linearly independent annihilators in Section 3.2. Further, in Section 3.3 we present algebraic immunity of a Boolean function in

terms algebraic immunity of its sub functions. Finally in Section 3.4 we study the algebraic immunity of certain classes of Boolean functions. The materials of this chapter are mainly based on [62].

In Chapter 4 we present the first ever construction to generate Boolean functions on any number of variables having optimal algebraic immunity. Then we study some other cryptographic properties of the constructed functions. For this chapter the materials are obtained from [63]. A revised version of [62, 63] appears in [36].

In Chapter 5 we present the basic theory to construct Boolean functions having optimal algebraic immunity. Using this theory we present symmetric (and also non symmetric) Boolean functions with optimal algebraic immunity. We study the other cryptographic properties for the construction in certain cases. This chapter is based on [66].

In Chapter 6 we study the immunity of Boolean functions against fast algebraic attacks. We explore the behavior of Boolean functions having optimal algebraic immunity in terms of fast algebraic attack. In this context we study the performance of various classes of Boolean functions. In Section 6.5 we propose some constructions to provide balanced Boolean functions having optimal algebraic immunity with certain strength against fast algebraic attack. For this chapter we have taken materials from [64].

To implement algebraic attack, finding low degree annihilators is essential. To find annihilators, generally one needs to solve a set of homogeneous linear equations. In Chapter 7 we present a strategy to reduce the size of the system of homogeneous linear equations that in turn reduces time complexity to find the annihilators. Moreover, we use a heuristic to find an affine transformation which reduces the size of the system of equations further. For this chapter the materials are taken from [65].

We conclude this thesis in Chapter 8 with a summary of our work and related open problems.

1.2 Prerequisites

It is assumed that the reader is familiar with undergraduate level combinatorics, abstract algebra, linear algebra and concept of Boolean functions. The reader is referred to [96, 144, 110, 129] for basic materials in linear algebra, combinatorics, finite fields and Boolean circuits respectively. It is also required to have the basic knowledge on cryptography [167] and related combinatorial properties of the Boolean functions [90, 113].

Chapter 2

Background

We start this chapter with relevant definitions of Boolean functions that are important in cryptography. Since we study some properties of Boolean functions and sequence of bits, our base field is the binary field, i.e., Galois field of two elements denoted by $\mathbb{F}_2 = GF(2) = \{0, 1\}$. We refer the elements of $\mathbb{F}_2$ as bits. The field operations are addition ($+$), i.e., logical XOR of two bits and multiplication ($*$), i.e. logical AND ($\wedge$) of two bits (abusing the notation we generally use ab for $a*b$ for $a, b \in \mathbb{F}_2$). One may refer [110] for detailed materials on finite fields. We need to consider an n-dimensional vector space $(\mathbb{F}_2^n, +, \cdot)$ over the field $\mathbb{F}_2$, where $+$ and $\cdot$ are usual vector addition and scalar multiplication. In short, we can say $\mathbb{F}_2^n$ is the set of all n-tuples from $\mathbb{F}_2$. Abusing the notation we use the same symbol $+$ for vector addition, field addition and usual arithmetic addition. We define the inner product of two vectors $u = (u_1, u_2, \ldots, u_n), v = (v_1, v_2, \ldots, v_n) \in \mathbb{F}_2^n$ as $\langle u, v\rangle = u_1v_1 + u_2v_2 + \ldots + u_nv_n$. The complement of a bit b is the additive inverse of b, i.e., $1 + b$ and denoted as $\overline{b}$. The complement of a vector v is the bitwise complement of each bit and denoted as $\overline{v}$. A string of bits $s = s_1s_2 \ldots s_l,\ s_i \in \mathbb{F}_2$ of length l can be viewed as an l-dimensional vector $(s_1, s_2, \ldots, s_l)$. Thus, we can apply the notations and definitions defined for vectors on string of bits as well.

Definition 1 *The (Hamming) distance between two vectors $u, v \in \mathbb{F}_2^n$ is the number of components where they are bitwise different. This is denoted as $d(u, v)$. The (Hamming) weight of a vector u is the number of nonzero components in the vector denoted as $\mathsf{wt}(v)$.*

Note that $d(u, v) = \mathsf{wt}(u + v)$ for $u, v \in \mathbb{F}_2^n$. The elements of $\mathbb{F}_2^n$ possesses a natural total ordering ($<_l$) known as the lexicographic ordering. For $u = (u_1, u_2, \ldots, u_n), v = (v_1, v_2, \ldots, v_n) \in \mathbb{F}_2^n,\ u <_l v$ if there is a $k,\ 1 \leq k \leq n$ such that $u_n = v_n,\ u_{n-1} = v_{n-1},\ \ldots, u_{k+1} = v_{k+1};\ u_k < v_k$, i.e., $u_k = 0$ and $v_k = 1$. So, one can order the vectors of

x_1	x_2	x_3	x_4	f
0	0	0	0	0
1	0	0	0	1
0	1	0	0	0
1	1	0	0	0
0	0	1	0	1
1	0	1	0	0
0	1	1	0	0
1	1	1	0	1
0	0	0	1	1
1	0	0	1	1
0	1	0	1	0
1	1	0	1	1
0	0	1	1	0
1	0	1	1	1
0	1	1	1	0
1	1	1	1	1

Table 2.1: Truth table representation

$\mathbb{F}_2^n$ as $\alpha_0 = (0, 0, \ldots, 0), \alpha_1 = (1, 0, \ldots, 0), \ldots, \alpha_{2^n-1} = (1, 1, \ldots, 1)$ such that $\alpha_0 <_l \alpha_1 <_l \ldots <_l \alpha_{2^n-1}$. We have bijective map from $\mathbb{F}_2^n$ to $\mathbb{Z}_{2^n}$, the ring of integers modulo 2^n as $v = (v_1, v_2, \ldots, v_n) \in \mathbb{F}_2^n$ maps to the integer $v_1 \cdot 2^0 + v_2 \cdot 2^1 + \ldots + v_n \cdot 2^{n-1}$. With this correspondence, the natural ordering of integers coincides with the earlier ordering $<_l$ of vectors. For easy computation and understanding, sometimes we write the integers to mean the corresponding vectors.

2.1 Boolean Functions

An $\mathbb{F}_2$ valued (i.e., the range set) function f on $\mathbb{F}_2^n$ (i.e., the domain set), i.e., $f : \mathbb{F}_2^n \mapsto \mathbb{F}_2$ is called an n-variable Boolean function. The set of all n-variable Boolean functions is denoted as $\mathcal{B}_n$. Unless stated otherwise, by a function, we shall mean a Boolean function. One may refer to [90, 113] for relevant works in the area of cryptographically significant Boolean functions.

2.1.1 Representation of Boolean Functions

Though there are many ways to represent a Boolean function [121], we will mainly concentrate on two ways that are frequently used to represent cryptographically significant Boolean functions. One way is to represent the Boolean function by a binary string called the *truth table* (TT) and the other way is to represent it as a multivariate polynomial known as the *algebraic normal form* (ANF).

Truth Table Representation

Very often we represent a Boolean function by the output column of its truth value according to the natural order ($<_l$) of input vectors from vector space $\mathbb{F}_2^n$. See Table 2.1 for an example of the truth table of a 4-variable function. Since the truth values are placed according to a total ordering of the input vectors, we can represent uniquely all the truth values by a binary string ($\mathcal{T}_f$) of length 2^n as following:

$$\mathcal{T}_f = f(\alpha_0)\ f(\alpha_1)\ \ldots\ f(\alpha_{2^n-1}),$$

where $\alpha_i,\ 0 \leq i \leq 2^n-1$ are vectors from $\mathbb{F}_2^n$ with ordering $<_l$. The truth table representation of the function in Table 2.1 is 0100100111010101.

Since the truth table of an n-variable Boolean function can be viewed as a vector of dimension 2^n, the set $\mathcal{B}_n$ forms a vector space $\mathbb{F}_2^{2^n}$ of dimension 2^n over $\mathbb{F}_2$. There are 2^{2^n} many distinct n-variable Boolean functions. So, we can use the same definitions for weight of a function f and distance between two functions f, g as explained for vectors of $\mathbb{F}_2^n$ in Definition 1.

Definition 2 *The* support *of a Boolean function $f \in \mathcal{B}_n$ is defined as $supp(f) = \{v \in \mathbb{F}_2^n \mid f(v) = 1\}$.*

So, the weight of a function $f \in \mathcal{B}_n$ is $\mathsf{wt}(f) = |supp(f)|$.

Definition 3 *A function $f \in \mathcal{B}_n$ is said to be* balanced *if its output column in the truth table contains equal number of 0's and 1's (i.e. $\mathsf{wt}(f) = 2^{n-1}$).*

It is easy to see that there are $\binom{2^n}{2^{n-1}}$ many balanced functions in the set of all n-variable Boolean functions. Note that the combining function in any cryptographic system need to be balanced.

ANF Representation

Another way of representing an n-variable Boolean function is in the polynomial form over the field $\mathbb{F}_2$ with n many indeterminates $x_1, x_2, \ldots, x_n$. Hence, it can be uniquely (up to permutation of indeterminates and monomials) represented in the ring $\mathbb{F}_2[x_1, x_2, \ldots, x_n]/\langle x_1^2 - x_1, x_2^2 - x_2, \ldots, x_n^2 - x_n\rangle$ as follows:

$$\begin{aligned} f(x_1, x_2 \ldots, x_n) &= a_0 + \sum_{i=0}^{n} a_i x_i + \sum_{1 \le i < j \le n} a_{i,j} x_i x_j + \cdots + \\ &\sum_{1 \le i_1 \le \cdots \le i_{n-1} \le n} a_{i_1,\ldots,i_{n-1}} x_{i_1} \ldots x_{i_{n-1}} + a_{1,\ldots,n} x_1 \ldots x_n, \end{aligned} \tag{2.1}$$

where $a_0, a_1, \ldots, a_{1,\ldots,n} \in \mathbb{F}_2$ are called the coefficients of the respective monomials. Equation 2.1 can be written in another way as

$$f(x_1, x_2, \ldots, x_n) = \sum_{(v_1, v_2, \ldots, v_n) \in \mathbb{F}_2^n} A_f(v_1, v_2, \ldots, v_n) x_1^{v_1} x_2^{v_2} \ldots x_n^{v_n} \tag{2.2}$$

where $A_f(v_1, v_2, \ldots, v_n)$ is again a Boolean function. This representation is called the *algebraic normal form (ANF)* of f.

Definition 4 *The* algebraic degree *or, simply the* degree *of the function f is defined as* $\deg(f) = max\{\mathsf{wt}(v) \mid A_f(v) = 1, v \in \mathbb{F}_2^n\}$.

The maximum algebraic degree achievable for an n-variable Boolean function is n. The degree of an n-variable Boolean function is n iff it is of odd weight. The maximum algebraic degree of an even weight n-variable function is at most $n - 1$.

Definition 5 *A Boolean function f is* affine *if there exists no term of degree greater that* 1 *in its ANF, i.e., $A_f(v) = 0$ for $\mathsf{wt}(v) > 1$ and the set of all n-variable affine functions is denoted as $\mathcal{A}_n$. Any affine function can be written as $l_{u,b}(x) = \langle u, x\rangle + b$, where $u = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$ and $b = a_0 \in \mathbb{F}_2$. An affine function with constant term equal to zero (i.e., $a_0 = 0$) is called a* linear *function and the set of all n-variable linear functions is denoted as $\mathcal{L}_n$.*

From a truth table of an n-variable Boolean function f, the ANF can be computed as

$$f(x_1, x_2, \ldots, x_n) = \sum_{(v_1,\ldots,v_n) \in supp(f)} (x_1 + v_1 + 1)(x_2 + v_2 + 1) \ldots (x_n + v_n + 1).$$

In other way, one can draw the truth table $\mathcal{T}_f$ from ANF of f as

$$\mathcal{T}_f = f(\alpha_0)\ f(\alpha_1)\ \ldots\ f(\alpha_{2^n-1}),$$

where the $\alpha_0, \alpha_1, \ldots, \alpha_{2^n-1} \in \mathbb{F}_2^n$ ordered in natural ordering ($<_l$).

Example 1 *ANF of Boolean function presented in Table 2.1 is* $x_1 + x_3 + x_4 + x_1x_2 + x_2x_3 + x_1x_2x_3 + x_2x_3x_4$ *and the degree of the Boolean function is* 3.

Product of two Boolean functions $f, g \in \mathcal{B}_n$ is denoted as $f * g$ (for simplicity sometimes we denote as fg instead of $f * g$). The truth table representation of fg is the point wise multiplication (logical AND) of the truth table string of f and g and ANF representation is the polynomial multiplication of f and g.

2.1.2 Walsh Transformation

Walsh transform is one of the most important tools to analyse a Boolean function. The Walsh transform of an n-variable Boolean function f is an integer valued function $W_f : \mathbb{F}_2^n \mapsto [-2^n, 2^n]$ defined as (see [112, page 414])

$$W_f(u) \ = \ \sum_{x \in \mathbb{F}_2^n} (-1)^{f(u)+\langle u,x \rangle}. \tag{2.3}$$

The term $W_f(u)$ is called the Walsh coefficient of f at the point u. The set of all the Walsh coefficients is referred as the *Walsh spectrum* of f. The conservation law for the spectral values of f is known as *Parseval's Theorem* (see [73]), which says that sum of the square of Walsh coefficients is constant, i.e., $\sum_{u \in F_2^n} W_f^2(u) = 2^{2n}$.

Example 2 *The Walsh spectra of Boolean function presented in Table 2.1 is presented in the following table.*

u	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
$W_f(u)$	0	8	−4	−4	0	0	−4	4	4	−4	0	0	4	4	0	8

Table 2.2: Walsh transform of the function in Table 2.1

2.1.3 Nonlinearity

Definition 6 *The nonlinearity* $\mathsf{nl}(f)$ *of an* n*-variable Boolean function is defined as*

$$\mathsf{nl}(f) = \min_{l \in \mathcal{A}_n} d(f, l).$$

Nonlinearity can be expressed in terms of Walsh spectra of f *as*

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

From cryptographic and coding theoretic studies, nonlinearity is a very important combinatorial property of Boolean functions. A function with low nonlinearity is prone to *Best Affine Approximation* (BAA) [73, Chapter 3] attack. It is a known plaintext attack and the attack needs the knowledge of the combining function. Best Affine Approximation means approximating the combining function by an affine function. Thus for cryptographic applications we need functions with high nonlinearity so that they can not be well approximated using the affine ones. Apart from its importance in cryptography, highly nonlinear Boolean functions are important combinatorial objects by themselves and have very close relationship with coding theory [112].

An n-variable (n even) function f achieves maximum nonlinearity iff $W_f(u) = \pm 2^{\frac{n}{2}}$, for all $u \in \mathbb{F}_2^n$ (using the Parseval's equality) and the nonlinearity is $2^{n-1} - 2^{\frac{n}{2}-1}$. Functions achieving this value of nonlinearity are called *bent* functions and they exist only when n is even [146]. The bent functions are unbalanced and for $n \geq 4$ the algebraic degree of a bent function is at most $\frac{n}{2}$. This class of functions are important in both cryptography and coding theory. For more details on bent functions, one may refer to [146, 112, 73, 31, 39]. The maximum possible nonlinearity for balanced functions on even number of variables is still open.

When n is odd, the maximum nonlinearity achievable by an n-variable Boolean function is not known (see [8, 139] for important results in this area). However, functions achieving a nonlinearity of $2^{n-1} - 2^{\frac{n-1}{2}}$ can be easily constructed [27] by concatenating two bent functions. It has been proved [8, 131, 95] that nonlinearity of n (odd) variable function is at most $2^{n-1} - 2^{\frac{n-1}{2}}$ for $n \leq 7$. On the other hand, Patterson and Wiedemann [139, 140] provided construction of functions with nonlinearity strictly greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd $n \geq 15$. Very recently, Kavut, Maitra and Yücel found functions of 9 variables with nonlinearity 241 and using these one can construct Boolean function with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd $n \geq 9$ [102].

2.1.4 Correlation Immunity and Resiliency

Definition 7 *An n-variable function f is called* correlation immune *of order t (t-CI) if $W_f(\omega) = 0$ for all ω with $1 \leq \mathsf{wt}(\omega) \leq t$ [161, 175]. A balanced t-CI function is called* t-resilient.

Note that a function is balanced if and only if $W_f(0) = 0$. Thus, a function f is t-resilient iff its Walsh transform satisfies $W_f(\omega) = 0$, for $0 \leq \mathsf{wt}(\omega) \leq t$. For t-CI (respectively, t-resilient) functions, the algebraic degree d is bounded by $d \leq n - t$ (respectively, $d \leq n - t - 1$) [161].

Example 3 *The Boolean function presented in Example 1 is not* 1*-CI as* $W_f(0001) = 8$ *(see Table 2.2), but it is balanced.*

An important attack called *divide-and-conquer* attack was proposed by Siegenthaler [162]. If the underlying function is not correlation immune (resilient) of certain order, then one can implement divide-and-conquer attack on the nonlinear combiner model. One may refer to [24, 155, 40, 127, 82, 115, 150] for detailed study and construction of correlation immune and resilient functions. Following the notation as in [150, 151, 165], we use (n, m, d, σ) to denote an n-variable, m-resilient function with degree d and nonlinearity σ. Further, by $[n, m, d, \sigma]$ we denote unbalanced n-variable, mth order correlation immune function with degree d and nonlinearity σ.

2.1.5 Symmetric and Rotation Symmetric Boolean Functions

We have already discussed in first chapter that a variety of criteria for choosing Boolean functions with cryptographic applications have been identified. It is very difficult (may not be possible also) to construct or find out a Boolean function which satisfies the optimality of all the properties. The trade-offs among these criteria have received a lot of attention in Boolean function literature for a long time (see [114] and references in this paper). It is difficult to search an appropriate functions from the whole set of Boolean functions as the search space is huge. Thus a natural idea is to decrease the search space by considering certain sub classes. Here we mention two such sub classes of functions.

Definition 8 *A Boolean function is called* symmetric *if it outputs the same value for all the inputs of same weight.*

Thus it is clear that one can represent an n-variable symmetric Boolean function $f(x_1, \ldots, x_n)$ in a reduced form by $n+1$ bits string re_f such that

$$re_f(i) = f(x_1, x_2, \ldots, x_n) \text{ when } wt(x_1, x_2, \ldots, x_n) = i.$$

It is also clear that in the algebraic normal form, a symmetric Boolean function will either contain all the terms of the same degree monomial or none of them. Thus we can represent the algebraic normal form in a reduced form by $n+1$ bits string ra_f such that $ra_f(i) = 1$, when all the i degree monomials are present and $ra_f(i) = 0$, when all the i degree monomials are absent.

Thus for an n-variable symmetric Boolean function f, both re_f, ra_f can be seen as mappings from $\{0, 1, \ldots, n\}$ to $\mathbb{F}_2$. One can follow [30, 152, 89] for details of symmetric functions. Now we define a larger sub class of Boolean functions.

Definition 9 *A Boolean function $f \in \mathcal{B}_n$ is called* rotation symmetric (RSBF) *if for each input $(x_1, \ldots, x_n) \in \mathbb{F}_2^n$, $f(\rho_n^k(x_1, \ldots, x_n)) = f(x_1, \ldots, x_n)$ for $1 \leq k \leq n$ where ρ_n^k acts as k-cyclic rotation on an n-bit vector, i.e., $\rho_n^k(x_1, x_2, \ldots, x_n) = (x_{1+k}, x_{2+k}, \ldots, x_n, x_1, \ldots, x_k)$.*

That is, the rotation symmetric Boolean functions are invariant under cyclic rotation of inputs. The set of RSBFs are interesting to look into as the space is much smaller ($\approx 2^{\frac{2^n}{n}}$) than the total space of Boolean functions (2^{2^n}). Recently the class of rotation symmetric Boolean functions (RSBFs) has received a lot of attention as the class contains functions with very good cryptographic properties [44, 61, 82, 94, 102, 122, 120, 141, 164, 165, 67]. The combinatorial analysis of such functions is also very interesting as they possess certain nice structures.

2.2 Algebraic Attack

Now we will briefly discuss a few specific algebraic attacks available in recent literature and following that we will explain the motivation of this thesis. The study in this area is mainly towards cryptanalysis of the symmetric key ciphers. However, algebraic attack is implementable over public key ciphers too. Several public key cryptosystems can be described by multivariate quadratic (MQ) equations, such as the cryptosystems of the hidden field equations (HFE) family [138]. Some works on algebraic attacks to recover the secret key of HFE have been presented in [104, 49, 79]. As RSA falls in Patarin's hidden field equations (HFE) [137], it may also be possible to analyse RSA in this direction. Symmetric key ciphers

have received more attention in terms of algebraic attack and next we will present a brief overview of algebraic attacks on both block and stream ciphers.

The basic principle of algebraic attacks comes from Shannon's work: "they consist in expressing the whole cryptosystem as a large system of multivariate algebraic equations which can be solved to recover the secret key" [159, page number 704]. Each algebraic equations can be viewed as a polynomial over the bits of the secret keys and equated to zero. In this kind of attack, the attacker finds a large set of multivariate equations over secret keys. Then the attempt is to solve this large set of multivariate equations to get the secret key or to reduce the search space. Primarily, this attack is known plaintext attack (some cases it is chosen plaintext attack) and algebraic in nature rather than statistical. The efficiency of this attack depends on the efficiency of the algorithm to generate and solve a large set of multivariate equations.

The algebraic attacks are implemented in two main steps :

1. Generating a large set of multivariate polynomial equations over secret keys.
2. Solving the system of generated equations to get the actual secret key or to reduce the search space for the key.

We discuss these two issues separately.

2.2.1 Generating the Multivariate Equations

The strategies of finding the algebraic equations depend on the internal structure of the cryptosystem. So, for different ciphers, attacker may follow different strategies to find relations. Here we discuss generating equations for block ciphers and stream ciphers in two different sub sections.

Block Ciphers

The security of block ciphers relies on the fact that the classical way of cryptanalysis, like linear and differential attacks which are probabilistic, becomes harder (complexity increases exponentially) as the number of rounds increases. Generally, the block ciphers (like DES, Rijndael, Serpent) use S-boxes in the design. For any S-box, one can generate a set of linearly independent multivariate equations between input and output bits. Multivariate

equations are said to be *linearly independent* if they are linearly independent when every distinct monomial is considered as a new variable.

In [80], Ferguson et. al. showed the complete AES can be represented by an equation with 2^{50} terms, which is too large to solve. Let us now consider the S-boxes of the form $s : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ which are multi output Boolean functions with high algebraic degree. That means we do not have low degree equations like $y_i = p_i(x_1, x_2, \ldots, x_n)$ for $1 \leq i \leq n$ where $(y_1, y_2, \ldots, y_n) = s(x_1, x_2, \ldots, x_n)$. However it does not guarantee that there is no low degree equation of the form $P(x_1, \ldots, x_n, y_1, \ldots, y_n) = 0$. The weak internal algebraic structure of the S-boxes may provide low degree algebraic equations which can be exploited for algebraic attacks.

In [58], Courtois and Pieprzyk attempted to analyse the block ciphers Rijndael and Serpent in terms of algebraic attacks and they could express the S-boxes of Serpent and Rijndael using algebraic equations of low degree d, e.g., $d \leq 2$. The authors found that there are at least 21 many quadratic equations for the 4-bit S-boxes used in Serpent. For the 8-bit S-box in Rijndael, the paper [58] reported 39 many quadratic equations with probability 1 and total 137 many monomials were present in the equations.

In [130], Murphy and Robshaw studied the algebraic structure of BES (a modified version of AES). From this they could identify that AES encryption can be described by a sparse system of multivariate quadratic equations over $GF(2^8)$.

In [11], Biryukov and Cannière have constructed a systems of equations (linear and quadratic) over $GF(2)$ and $GF(2^8)$ for the 128-bit key block ciphers KHAZAD, MISTY1, KASUMI, CAMELLIA, RIJNDAEL and SERPENT and computed some properties that might influence the complexity of solving them to recover the key. In [11, Table 3] of the paper, they compare the complexities of the ciphers over $GF(2)$ in terms of number of equations and monomials. Then in Table 4, they compare the complexities of CAMELLIA-128, RIJNDAEL-128 over $GF(2^8)$.

Power functions are frequently used in constructing the S-boxes in block ciphers. We refer a power function as $X \mapsto X^\alpha$ over finite field $GF(2^n)$. These functions are classified according the value of α and some of them are used for designing the ciphers. For example, in AES, the inverse function $X \mapsto X^{-1} = X^{2^n-2}$ is used. In [54], the authors could generalize the count of quadratic equations in [58] for the n-bit S-box in Rijndael; the count is $5n-1$ many quadratic equations. Further, in [54] some error from the paper [41] in the count of quadratic equations for Gold exponents [85] (which is a mapping $X \mapsto X^{2^k+1}$ with $\gcd(k, n) = 1$) is identified. In [54, Table 2], the number of bi-affine and quadratic equations for $n =$

$2, 3, 4, 5, 7, 8, 9, 15, 16, 17$ with $k = 1, 2, 3, 4$ is tabulated. The authors of [54] also studied

1. the Dobbertin exponents [75] of the form $X \mapsto X^{2^{4k}+2^{3k}+2^{2k}+2^k-1}$ with $n = 5k$,
2. Niho exponents [74] of the form $X \mapsto X^{2^k+2^{k/2}-1}$ with $n = 2k + 1$ and k even or, $X \mapsto X^{2^k+2^{(3k+1)/2}-1}$ with $n = 2k + 1$ and k odd,
3. Welch exponents [75] of the form $X \mapsto X^{2^k+3}$ with $n = 2k + 1$ and
4. Kasami exponents [101] of the form $X \mapsto X^{2^{2k}-2^k+1}$ with $\gcd(n, k) = 1$ and $1 \leq k \leq n/2$.

In [134], the authors study the algebraic structures of the component functions (note that each component function of an S-box is a Boolean function) of power functions like inverse, Kasami and Niho functions.

Stream Ciphers

Stream ciphers are potentially vulnerable to algebraic attacks and recently good amount of research effort has been put into this area. In this section we outline the existing literature on how to generate the algebraic equations of low degree. The two basic models of LFSR based stream cipher for generating keystream are nonlinear combiner and nonlinear filter generator. *Nonlinear combination* generator uses a number of LFSRs and their outputs are fed to a nonlinear function to generate the keystream. In this case the number of variables of the nonlinear function is equal to the number of LFSRs. For *nonlinear filter* generator, the output is computed by a nonlinear function taking the values from some taps of a single LFSR. We discuss the strategies to generate equations in unified way for both the models.

Suppose the model uses k-bit state LFSRs and each time the state is modified by a linear update function L. Let the initial state be $S^0 = (s_0, s_1, \ldots, s_{k-1})$. At the t-th clock the output of the keystream will be $z_t = f(S^t), t \geq 0$, where f is the nonlinear function; $S^t = L^t(S^0)$ denotes the state when the linear function L is operated t times on the state S^0. The problem is to recover the initial state $S^0 = (s_0, s_1, \ldots, s_{k-1})$. Here one can exploit the known plaintext attack where some (say l many) plaintext bits and corresponding cipher text bits are known and XORing them the keystream bits are found. Knowing some keystream bits (say, $z_{k_1}, z_{k_2}, \ldots, z_{k_l}$) one can generate a system of equations of degree equal to $\deg(f)$

as follows:

$$\begin{aligned} f(L^{k_1}(S^0)) &= z_{k_1}, \\ f(L^{k_2}(S^0)) &= z_{k_2}, \\ &\vdots \ \ \vdots \ \ \vdots \\ f(L^{k_l}(S^0)) &= z_{k_l}. \end{aligned} \tag{2.4}$$

The complexity to solve this system of equations increases if the degree of the nonlinear function f is high (in general, the high degree function is used to generate keystream sequence with good linear complexity). So, generating equations in this way may not be efficient for algebraic attack. One may like to generate low degree equations using some weaknesses in the internal structure of nonlinear functions. Towards this we refer [56, 123], where the authors used low degree multiples and annihilators of the nonlinear function to generate the low degree equations. In [56], the low degree multiples of the nonlinear function f are exploited for algebraic attack as follows. At the time t, the output bit z_t gives the equation $f(L^t(S^0)) = f(S^t) = z_t$. The main idea consists of multiplying $f(S^t)$ (that is usually of high degree) with a well chosen function $g(S^t)$, such that the degree of fg is substantially reduced. If $z_t = 0$ then we get the equation $f(S^t)g(S^t) = h(S^t)$, i.e., $z_t g(S^t) = h(S^t)$ which implies $h(S^t) = 0$. So, we get equations of low degree if degree of h is low. Using this technique the authors identified how to reduce the complexity of the attack. Further in [123], authors extended this idea to generate equations using annihilators. Here we define some terms in this context.

Definition 10 *Given $f \in \mathcal{B}_n$, a nonzero function $g \in \mathcal{B}_n$ is called an annihilator of f if $f(x)g(x) = 0$ for all $x \in \mathbb{F}_2^n$. We also define the following sets:*

1. *$AN(f) = \{g \in \mathcal{B}_n \mid g$ nonzero, $fg = 0\}$, i.e., set of all annihilators of f.*
2. *$AN_{\leq d}(f) = \{g \in \mathcal{B}_n \mid g$ nonzero, $\deg(g) \leq d$, $fg = 0\}$, i.e., set of all annihilators of degree $\leq d$ of f.*
3. *$AN_d(f) = \{g \in \mathcal{B}_n \mid g$ nonzero, $\deg(g) = d$, $fg = 0\}$, i.e., set of all d degree annihilators of f.*

Now consider the following two scenarios to generate equations for the Boolean function f which is usually of high degree.

1. Assume that there exists a function g of low degree such that the function $h = fg$ is of low degree and h is nonzero.

2. Assume there exists a nonzero function g of low degree such that $fg = 0$, i.e., existence of low degree annihilator g of f.

Without loss of generality, in scenario 1 above, one may consider that $\deg(g) \leq \deg(h)$. This is because, if $\deg(g) > \deg(h)$, then $fh = ffg = fg = h$, so one can use h in place of g. In this context, the result from [56] on the bounds of the degree of g and h is as follows.

Theorem 1 *[56] For any $f \in \mathcal{B}_n$, there is a nonzero $g \in \mathcal{B}_n$ of degree at most $\lceil \frac{n}{2} \rceil$ such that fg is of degree at most $\lceil \frac{n}{2} \rceil$.*

Now we consider the scenario 1. Here $fg = h$, $g \neq 0$. Now, $f(g + h) = fg + fh = h + h = 0$. So, if $g \neq h$, we have $g + h \in AN(f)$ which comes under scenario 2. Then $(1 + f)h = h + fh = h + h = 0$. So, $h \in AN(1 + f)$. So, we can redefine the above scenarios as following:

A. Assume there exists a function g of low degree such that $fg = 0$, i.e., we consider the existence of a low degree annihilator g of f.

B. Assume there exists a function h of low degree such that $(1 + f)h = 0$, i.e., we consider the existence of a low degree annihilator h of $1 + f$.

Corollary 1 *[123] For any $f \in \mathcal{B}_n$, there exists a nonzero $g \in \mathcal{B}_n$ of degree at most $\lceil \frac{n}{2} \rceil$ such that $fg = 0$ or $(1 + f)g = 0$.*

In [123] these two scenarios are exploited to generate equations of low degree. Suppose one finds n_1 (respectively n_2) many linearly independent annihilators g_i, $1 \leq i \leq n_1$ (respectively h_i, $1 \leq i \leq n_2$) of n variable function f (respectively $1 + f$) having degree less than or equal to d. For the attack, it is assumed that the linear feedback connections are known. Further we assume that some of the output keystream bits are known. So the output bit z_t at the time t generates an equation $f(L^t(S^0)) = z_t$. If the bit z_t is 1, one may consider scenario A, i.e., $fg_i = 0$ to get the equation $f(L^t(S^0))g_i(L^t(S^0)) = z_t g_i(L^t(S^0))$, i.e., $g_i(L^t(S^0)) = 0$ for $1 \leq i \leq n_1$. So, one can get n_1 many equations for each known bit $z_t = 1$. Similarly if the bit z_t is 0, one may consider scenario B, i.e., $(1 + f)h = 0$ to get the equation $(1 + f(L^t(S^0)))h(L^t(S^0)) = z_t h(L^t(S^0))$, i.e., $h(L^t(S^0)) = 0$ for $1 \leq i \leq n_2$. So, one can

get n_2 many equations for each known bit $z_t = 0$. If one knows l_0 and l_1 many output bits which are 0's and 1's respectively, then $l_0 n_2 + l_1 n_1$ many equations (all may not be linearly independent) of degree less than or equal to d can be generated.

In general, the efficiency of solving the system of multivariate equations depends on the degree of the equations. That is, a system of multivariate equations of lower degree can be solved more efficiently than a system of higher degree. So, if a function f or its complement $1 + f$ has low degree annihilators then one can generate low degree equations. Towards this argument we define algebraic immunity of a Boolean function.

Definition 11 *[123] Given $f \in \mathcal{B}_n$, its algebraic immunity is defined as the minimum degree of all nonzero annihilators of f or $1 + f$, and it is denoted by $\mathsf{AI}_n(f)$. That is $\mathsf{AI}_n(f) = \min\{\deg(g) \in \mathcal{B}_n \mid g \neq 0,\ g \in AN(f) \cup AN(1 + f)\}$.*

At this point we like to discuss whether the term "algebraic immunity" of a Boolean function is appropriate. Recently there are many works in the area of algebraic attacks and some of the initial and important papers are [58, 56, 123]. It is now clear that a Boolean function or its complement, used in a cryptosystem, should not have low degree annihilators. However, the algebraic normal form (ANF) of the annihilators are also important. It may very well happen that an annihilator with higher degree may have a few terms and on the other hand an annihilator with lower degree may have many more terms in the ANF and in certain cases, it may be better to use the high degree annihilator with fewer terms than the low degree annihilator with more terms for the algebraic attack. Thus increase in the degree of annihilator (of the Boolean function) may not be the only measure in terms of resistance of a cryptosystem (that uses the Boolean function) against algebraic attack. Moreover, recently observed fast algebraic attacks [51, 3, 53, 20] are also very effective for cryptanalysis (see Section 2.2.3). In fast algebraic attacks one may not need low degree annihilators to implement the attack. Based on the existing research so far, it is difficult to formalize or quantify the measure of resistance of a Boolean function used in a cryptosystem against algebraic or fast algebraic attacks. It clearly depends on how the Boolean function is used in the construction of cryptosystem and how the algebraic attack is designed against the complete scheme.

On the other hand, if one just concentrates on a Boolean function, then it is meaningful to consider the annihilators of $f, 1+f$ to study its resistance against algebraic attack and one would always like to get a Boolean function f, such that both f and $1 + f$ do not have any annihilator with degree less than $\lceil \frac{n}{2} \rceil$. Further, if one considers the algebraic degree of an n-variable Boolean function, then it may very well happen that the function $f(x_1, x_2, \ldots, x_n)$

is of very good algebraic degree, but if one conditions one variable, say $f(x_1 = 0, x_2, \ldots, x_n)$, the degree falls drastically. However, this is not true in terms of algebraic immunity. It can be checked that if f has algebraic immunity t, then after conditioning any k variables, the algebraic immunity of the sub function on $n - k$ variables will be $\geq t - k$. This is clearly a stronger property than the algebraic degree of a Boolean function. Based on these arguments and as the term has already been appeared in many papers [16, 17, 18, 123, 33], one may be tempted to use the term "algebraic immunity".

In one of our papers [66] we use the term "annihilator immunity" instead of "algebraic immunity" as this clearly quantifies the measure how good a Boolean function is in terms of not having low degree annihilators. However, in the current literature (except [66]) algebraic immunity is well accepted to define this property. Hence, in this thesis we use the term "algebraic immunity" (AI) itself.

2.2.2 Solving the System of Multivariate Equations

Since a cipher can be described by algebraic multivariate equations, one may attack the cipher if the system of equations can be solved using feasible resources. For example, in [58] it has been pointed out that one can recover an AES-128 key with a high probability from one AES-128 plaintext-ciphertext pair if one can solve certain systems with 1600 variables and 8000 quadratic equations over $\mathbb{F}_2$, and it has been pointed out in [130] that one can achieve the same goal if one can solve certain systems with 3986 variables and 3840 (sparse) quadratic equations as well as 1408 linear equations over $\mathbb{F}_{2^8}$. Solving the system of multivariate algebraic equations is an important area in the field of computational algebraic geometry and commutative algebra. This problem is NP-complete [84] even if all the equations are quadratic and base field is $\mathbb{F}_2$ [49, Section 3]. As both time and space complexities of solving the system are important in cryptanalysis, we refer to some existing techniques to solve the system of multivariate equations in cryptographic aspect. We will briefly discuss these algorithms, e.g., linearization, XL, XSL, Gröbner bases algorithms like Buchberger algorithms, F4 and F5. One may refer [25] for more details.

Problem 1 *Let the base field is $GF(q) = \mathbb{F}_q$ and $e_1, e_2, \ldots, e_m$ are m multivariate polynomials with n variables over the field $\mathbb{F}_q$ with $X_i^q = X_i$ for $1 \leq i \leq n$, i.e., $e_j \in \mathbb{F}_q[X_1, X_2, \ldots, X_n]/\langle X_1^q = X_1, \ldots, X_n^q = X_n\rangle$ for $1 \leq j \leq m$. The problem is to find the solution(s) $(x_1, x_2, \ldots, x_n) \in \mathbb{F}_q^n$ such that $e_i(x_1, x_2, \ldots, x_n) = 0$ for $1 \leq i \leq m$.*

In our discussions we always consider $q = 2$ unless specified otherwise.

Linearization

The method of *linearization* is the simplest and most popular technique for solving the system of multivariate polynomial equations. In this technique, each monomial is considered as an independent indeterminate and hence the total system can be rewritten as a system of linear equations with a large number of indeterminates (in fact, the number of new variables is equal to the number of monomials involved in the older system of multivariate polynomial equations). Formally, each polynomial e_i can be viewed as $e_i = \sum_{\alpha \in \mathbb{F}_2^n} c_\alpha^i X^\alpha$ where $c_\alpha^i \in \mathbb{F}_2$ and $X^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ for $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$. In this case each X^α is considered as a new variable if there is a nonzero c_α^i for $1 \leq i \leq m$. If the degree of equations are less than or equal to d then total number of new variables $M \leq \sum_{i=1}^{i=d} \binom{n}{i}$ (excluding the constant term).

Then the new system of linear equations can be solved in $O(M^\omega)$ time, ω being the exponent depending on the technique used. Using the well known Gaussian elimination technique we have $\omega = 3$; Strassen's algorithm [168] takes $\omega = \log_2 7 \approx 2.807$ and also the one by Coppersmith and Winogard in [48] takes $\omega = 2.376$. If the matrix is very sparse (number of nonzero entries are very few) then the complexity may be reduced to $O(M^2)$ [174].

In order to apply the linearization method, the number of linearly independent equations in the system needs to be approximately the same as the number of indeterminates of the system. When this is not the case, a number of techniques have been proposed that attempt to generate enough number of linearly independent equations. Now we discuss some of them.

Relinearization

In linearization method we may not always get a unique solution of a system of multivariate equations. There may be more than one solutions and some of them may be conflicting to each other as we consider a monomial as a separate variable. In Crypto 99, Kipnis and Shamir [104] introduced a new method (extending the linearization method) for solving overdefined system of quadratic polynomial equations, called *relinearization*. The general idea of this method is to use linearization technique to solve m equations in the $n(n+1)/2$ variables (i.e., number of 2 or less degree monomials on n variables). If the above discussed problem occurs then one can create some more equations exploiting the commutativity of the multiplication of x_i's. For example, $x_a x_b x_c x_d$ can be parenthesized in 3 different ways as follows: $(x_a x_b)(x_c x_d) = (x_a x_c)(x_b x_d) = (x_a x_d)(x_b x_c)$, i.e., writing in the form of new variables to get 2 more linearly independent equations $y_{ab} y_{cd} = y_{ac} y_{bd}$, $y_{ab} y_{cd} = y_{ad} y_{bc}$ where

$y_{ab} = x_a x_b$ etc. So, by this way one can generate larger number of quadratic equations. Then this system can be solved by linearization or recursive relinearization. Though the relinearization technique can solve many systems of equations which could not be solved by linearization, its exact complexity and success rate are still not well understood. In [104, Appendix A], a toy example is given for better understanding of the method. Further, in [55] Courtois et. al. analysed some theoretical and practical aspects of this relinearization technique.

Extended Linearization and its Variants

In Eurocrypt 2000 [55], Courtois et. al. showed that many of the equations generated by relinearization are actually linearly dependent, and hence relinearization is less efficient than one could expect. Then the authors [55] proposed an improved algorithm called XL (stands for *eXteneded Linearization* or, multiplication(X) and Linearization) which is considered to be simpler and more powerful than relinearization. In [55], the authors study the randomly chosen quadratic equations over a finite field. The basic idea of this technique is to generate, from each polynomial equation, a large number of higher degree variants by multiplying it with all the possible monomials of some bounded degree, and then to linearize the expanded system.

Let $e_1, e_2, \ldots, e_m$ be the equations in n variables and D be the parameter of the algorithm such that a system of linear equations of dimension $\sum_{i=0}^{D} \binom{n}{D}$ can be solved with in the feasible resources. This algorithm generates some more equations $X^\alpha e_j$ for $1 \leq j \leq m$ and $\alpha \in \mathbb{F}_2^n$ such that $\deg(X^\alpha e_j) \leq D$. Initially using linearization technique one can generate the system of linear equations and then solve the system (by Gaussian elimination) such that one variable terms (say, x_1) are eliminated towards the end. It is expected that one can land to at least one univariate equation (i.e., powers of x_1). This univariate equation can be efficiently solved over the finite fields (e.g., using Berlekamp's algorithm [125, Chapter 3.11]). One can then simplify the system by substituting the value (of x_1) and the process gets repeated for the other indeterminates. The algorithm is as follows:

Algorithm 1 *[55] (***The XL Algorithm***) Execute the following steps.*

1. **Multiply:** *Generate all the products $X^\alpha e_i$ with $\mathsf{wt}(\alpha) \leq D - 2$.*
2. **Linearize:** *Consider each monomial in x_i of degree $\leq D$ as a new variable and perform Gaussian elimination on the equations obtained in step 1. The ordering on the mono-*

mials must be such that all the terms containing one variable (say x_1) are eliminated last.

3. **Solve:** *Assume that step 2 yields at least one univariate equation in the power of x_1. Solve this equation over finite fields.*

4. **Repeat:** *Simplify the equations and repeat the process to find the values of the other variables.*

In [55], an improved variant of XL called FXL is proposed. It consists of guessing the values of a few variables and then applying XL. It seems that, for a system of multivariate quadratic (MQ) equations over a small field, the FXL might be subexponential. Later in [57], Courtois and Patarin proposed two more variants of XL called XL$'$ and XL2 by modifying the last step of the XL algorithm. In XL$'$, at the last step, one needs to search at least r many equations with only r many variables (in XL we need to obtain at least one equation in only one variable). It is expected that such a system will have one solution and it can be solved by exhaustive search over the finite field. Then substituting these r values, compute for other variables using Gaussian reduction. In XL2, the idea is to obtain at least one equation with a restricted set of variables, and from these equations to obtain new equations that were not in the original system. These equations will be added with the original system to form a new set of equations and this process will be iterated. By this way one may generate some more missing equations. Using this variants some more systems can be solved which XL can not. However in spite of computer simulations, it is not clear what is the exact time complexity of XL and its variants.

Though XL is simple, it is not clear for which n (the number of variables) and m (the number of equations) it terminates successfully and what is its asymptotic complexity. Particularly, this method is used to solve system of quadratic equations. So, this method and its variants are very popular for cryptanalysis of block ciphers and public key cryptosystems. In [55, 58, 57], the authors analysed for random quadratic equations and in [58], the authors implemented the method to cryptanalyse Rijndael, though it was not efficient. In [128, 176, 70], the complexity of XL have been studied. Diem has studied [70] the complexity of XL algorithm using Hilbert theory and a conjecture in commutative algebra and further presented some critical remarks regarding the complexity of the XL algorithm [55]. It seems that for a random system of quadratic equations over $\mathbb{F}_2$ that has a solution, XL method has very high chance to work (but for some special systems it always fails [128]). It sometimes fails because of the missing equations.

Courtois and Pieprzyk [58] have tried to analyse the Rijndael using XL algorithm. However, they found that for 256 bit cipher the complexity is about 2^{330} using the best Gaussian reduction exponent [48]. Clearly this is inefficient. They proposed an improved version of the XL algorithm which takes advantage of the sparsity and specific structure of the equations. Here, they suggest to multiply the equations by carefully selected monomials instead of multiplying all monomials of degree less than or equal to $D-2$. So, this process is named as XSL which stands for "eXtended Sparse Linearization" or "multiply(X) by Selected monomials and Linearize". To improve XSL one may use techniques used in FXL, XL$'$ and XL2 [55, 57]. Following the present literature it is evident that there are number of issues to be settled related to success and implementation of XL and its variants.

Gröbner Bases Algorithms

We now briefly discuss how to solve a system of multivariate equations by using the Gröbner bases for the ideal generated by the multivariate polynomials. One chooses the Gröbner bases which are in very simple form and then these are used to obtain the solution of the system. One may refer [59] for detailed study on this topic. More improved and recent algorithms related to this area are F_4 [77] and F_5 [78].

The Problem 1 is studied in abstract algebra in the following way. Let $I \subseteq \mathbb{F}_q[X_1, \ldots, X_n]$ be the ideal generated by the polynomials $e_1, \ldots, e_m$. It is denoted by $I = \langle e_1, \ldots, e_m \rangle$. The set of solutions

$$V(e_1, \ldots, e_m) = \{(x_1, \ldots, x_n) \in \mathbb{F}_q^n \mid e_i(x_1, \ldots, x_n) = 0 \text{ for all } 1 \leq i \leq m\},$$

is known as *affine variety* defined by $e_1, \ldots, e_m$. So, the problem is the same as asking for the points in the affine variety $V(e_1, \ldots, e_m)$. This problem can be solved using Gröbner bases of the ideal I. Here we discuss briefly on Gröbner bases of an ideal. Let $\preceq$ be a total monomial order on the set of monomials X^α, where $\alpha \in \mathbb{F}_q^n$. As the terms of each polynomial can be uniquely ordered with respect to $\preceq$, the notion of *leading coefficients* ($LC(f)$), *leading monomial* ($LM(f)$) and *leading terms* ($LT(f)$) are well defined.

Let $I \subseteq \mathbb{F}_q[X_1, \ldots, X_n]$ be an ideal and $LT(I) = \{LT(f) \mid f \in I\}$. A finite subset $G = \{g_1, \ldots, g_s\}$ of I is said to be a Gröbner basis if $\langle LT(g_1), \ldots, LT(g_s) \rangle = \langle LT(I) \rangle$. It can be shown that every nontrivial ideal has a Gröbner basis fixing a monomial order $\preceq$. It may not unique, but it is possible to get uniquely reduced Gröbner basis (see [59, Definition 5, Chapter2]). As the structure of the polynomials in Gröbner bases are very simple, it is easy to exploit G for finding $V(G)$. Since $V(G) = V(I) = V(e_1, \ldots, e_m)$, finding $V(G)$ is enough. So, the remaining task is to generate the Gröbner bases.

The Buchberger's algorithm [23] is the classical algorithm for computing the Gröbner bases. It uses the generalized Euclidean division algorithm to find out the remainder polynomial by dividing a polynomial by a set of polynomials. More precisely, given a monomial order, one can get the division algorithm $division(f, f_1, \ldots, f_l) = (h_1, \ldots, h_l, r)$, where $f = f_1h_1 + \ldots + f_lh_l + r$ and no $LM(f_i)$ divides any of the monomials of r. If the set $\{f_1, \ldots, f_l\}$ is a Gröbner basis of an ideal, then r is unique. The Gröbner basis of an ideal generated by $e_1, \ldots, e_m$ can be computed by the following Buchberger algorithm (see [59] for details).

Algorithm 2

Input: $E = (e_1, \ldots, e_m)$

Output: a Gröbner basis $G = (g_1, \ldots, g_s)$ *for* $I = \langle e_1, \ldots, e_m \rangle$.

Initialize: $G = E$;

Repeat

$G' = G$;

For each pair $\{p, q\},\ p \neq q$ *in* G' *Do*

Combine each $p,\ q$ *by canceling the leading terms to get* $S(p, q)$ *(the S-polynomial);*

Compute the remainders of $S(p, q)$ *by* G;

Augment the nonzero remainders with G;

Until $G = G'$;

It can be shown that the algorithm terminates and computes a Gröbner basis of the ideal $\langle e_1, \ldots, e_m \rangle$. The complexity of this algorithm is closely related to the total degree of the intermediate polynomials that are generated during the execution of the algorithm. In the worst case, it is known to run in doubly exponential time. For details one may refer to [59]. Regarding implementation, there are number of modifications that can be made to improve the performance of the algorithm. The F_4 [77] and F_5 (optimized version of F_4) [78] are well known improved versions of the Buchberger algorithm. The idea comes by viewing the Euclidean division algorithm in terms of matrix reduction algorithm. The implementation theory of F_4 is available in [77, 166] and the software is available at [171].

2.2.3 Fast Algebraic Attack

The idea of algebraic attack discussed so far may be successful if we have enough number of low degree equations and known keystream bits. Hence, the search for systems of low degree equations is a desirable goal in algebraic attacks. A clever idea is presented by Courtois in Crypto 2003 [51] to reduce the degree of equations in a precomputation step which is the heart of fast algebraic attack. Before solving the system of equations, all the higher degree monomials independent of the keystream bits are eliminated. The author used the equations of the form $h(K_t, z_t, z_{t+1}, \ldots, z_{t+r}) = 0$, where K_t is the input variable in terms of the states of the LFSRs at the t-th time and z_i is the i-th keystream bit that gives

$$h(K_t, z_t, z_{t+1}, \cdots, z_{t+r}) = u(K_t) + v(K_t, z_t, \ldots, z_{t+r}) = 0, \tag{2.5}$$

where u is of degree d in the bits of K_t, v is of degree $e < d$ in the bits K_t and only v depends on the keystream. After substitution of keystream bits $z_t, \ldots, z_{t+r}$ in Equation 2.5, we have equations like $u(K_t) = v(K_t)$, where degree of u and v are d and e respectively. Suppose the number of terms in u and v are $D \approx \sum_{i=0}^{d} \binom{n}{i}$ and $E \approx \sum_{i=0}^{e} \binom{n}{i}$ respectively. So the total number of monomials of system is order of D as D is much larger than E. When one wants to solve the equations the complexity will be $O(D^\omega) \approx O(n^d)$ where ω is the coefficient for solving system of equations. If one can eliminate the monomials of u then at the solving stage the time complexity will be of order $O(E^\omega) \approx O(n^e)$ which is smaller than the earlier one. To eliminate all the monomials of u, Courtios [51] proposed an algorithm using the Berlekamp-Massey algorithm [118, 125] where one can find a linear combination among the equations such that all the terms in u will be canceled out. The complexity for this precomputation phase requires $O(D^2)$ time using normal Berlekamp-Massey algorithm, while an asymptotically fast implementation has a complexity of $C \cdot D(\log D)$ for a large constant C. Unlikely the simple algebraic attack, one needs to know consecutive keystream bits to implement the first algebraic attack. So, fast algebraic attack is a chosen plain text attack.

The applicability of this algorithm was an open question as the proof of the algorithm was not provided in [51]. Later in FSE 2004 [3], the correctness of the algorithm was proved under the assumption that the period of LFSRs involved are co-prime. During the generation of equations one needs to substitute the key bits in the equations. The complexity for substituting the key bits was not calculated exactly in [51]. However, the simple substitution would require a complexity of $O(DE^2)$ which may dominate the overall complexity (see [93, Table 1]). Later in Crypto 2004, Hawkes and Rose [93] used Fast Fourier Transformation (FFT) [47] technique to reduce the time complexity at the substitution step

to $O(ED\log_2 D)$, which is smaller than the earlier method.

Moreover, in [93] the authors provide a more efficient algorithm for the precomputation step. The method requires order of $D(\log D)^2$ operations compared to the previously proposed strategy [51] that takes order of $C \cdot D(\log D)$ for a large C or D^2. More detailed comparisons are available in [93, Table 2 and Table 3].

The strategy used for algebraic attack [56] can be used to implement the fast algebraic attack on memoryless generators like filter and combiner generator [20]. In [20] (e, d)-relation of a Boolean function $f \in \mathcal{B}_n$ is defined. If there exist $g, h \in \mathcal{B}_n$ having degree e and d respectively, such that $fg = h$ then f has an (e, d)-relation. In equation finding step one can exploit the relation $f(x)g(x) = h(x)$ for all $x \in \mathbb{F}_2^n$ as

$$z_i g(K_i) = h(K_i), \tag{2.6}$$

where $z_i = f(K_i)$ is the i-th bit of keystream and K_i is the state of the LFSR at i-th clock. If $z_i = 0$ then the equation will be $h(K_i) = 0$ and if $z_i = 1$ then the equation will be $h(K_i) = g(K_i)$. Thus, knowing some z_i one can generate equations of degree $\max\{e, d\}$. In this case we need $d > e$, otherwise the equations will be the ones available from in the case of algebraic attack [56, 123]. Hence to describe the fast algebraic attack the (e, d)-relation will be interesting if $e < d$. In this system of equations the number of monomials is approximately $D = \sum_{i=0}^{d} \binom{n}{i}$. Hence to solve this system of equations one requires $O(D^\omega)$ complexity. Following the precomputation step in [51, 93], one can find a linear combination of the Equations 2.6 for $D + 1$ consecutive bits such that $\sum_{i=0}^{D} \alpha_i h(K_{r+i}) = 0$ for any r and $\alpha_i \in \mathbb{F}_2$. Now the Equations 2.6 can be reduced by eliminating the monomials of h as

$$\sum_{i=0}^{D} \alpha_i z_{r+i} g(K_{r+i}) = 0. \tag{2.7}$$

For different values of r one can generate different equations of degree e. For example, varying r from 0 to E one can have E many equations of degree e. Hence for this step one needs $D+E$ many consecutive keystream bits and $O(D(\log D)^2)$ time complexity [93]. Then in the keystream substitution step one may follow the FFT technique described in [93] with complexity $O(ED\log D)$. Since we have equations with E many monomials, one can solve the equations with complexity $O(E^\omega)$ using linearization technique. In [53] this kind of fast algebraic attack is used to cryptanalysis SFINKS [19].

2.2.4 Algebraic Attacks on some Stream Ciphers

Let us now briefly outline the algebraic attacks on a few specific stream ciphers.

Attack on Toyocrypt

Toyocrypt [126] was a submission to the Japanese government Cryptrec call for cryptographic primitives. It uses a 128-bit LFSR and a Boolean function of the form:

$$\begin{aligned} f(s_0,\ldots,s_{127}) &= s_{127}+\sum_{i=0}^{62} s_i s_{\alpha_i} + s_{10}s_{23}s_{32}s_{42} \\ &\quad + s_1 s_2 s_9 s_{12} s_{18} s_{20} s_{23} s_{25} s_{26} s_{28} s_{33} s_{38} s_{41} s_{42} s_{51} s_{53} s_{59} + \prod_{i=0}^{62} s_i, \end{aligned} \tag{2.8}$$

where $\{\alpha_0,\cdots,\alpha_{62}\}$ is a permutation of the set $\{63,\cdots,125\}$. This system is quite vulnerable to low degree approximation attack as the function in Equation 2.8 contain only two higher degree monomials (one of degree 63 and another one is of degree 17). A probabilistic algebraic attack with probability $1-2^{-17}$ and complexity 2^{92} CPU clocks has been presented in [50].

Then in Eurocrypt 2003, Courtois and Meier [56] could find out low degree multipliers of f which gives low degree annihilators of f and $1+f$. They found that the monomials of degree $4, 17$ and 63 (in fact, these are the higher degree monomials in f) contain the factor $s_{23}s_{42}$. So, $f*(1+s_{23})$ and $f*(1+s_{42})$ result two cubic polynomials. Using these two cubic equations, one may need $2^{17.4}$ keystream bits (need not be consecutive) to implement the algebraic attack by solving the equations by linearization technique. For that one needs 2^{49} CPU clocks, 16 GB memory and only 20 KB of (non-consecutive) keystream bits.

Attack on LILI-128

In designing of LILI-128 [163], a highly nonlinear 10-variable Boolean function of degree 6 has been used following [150]. Since the used function is of 10 variables, it must have annihilators of degree less than or equal to 5 (see Theorem 1 and Corollary 1). Thus, one can get equations of degree 5 or less which can be exploited to attack the cipher. There are 14 many 4 degree annihilators of f and $1+f$ that makes the attack faster. For this attack one needs 2^{57} CPU clocks, 762 GB memory and 2^{57} consecutive stream keystream bits. The details of this attack is available in [56].

Attack on SFINKS

SFINKS is a newly proposed stream cipher by Braeken et. al. [19] that has been submitted to ECRYPT call in April 2005. It uses a 256 state LFSR described by the formula

$$\begin{aligned} s_{t+256} &= s_{t+212} + s_{t+194} + s_{t+192} + s_{t+187} + s_{t+163} + s_{t+151} + s_{t+125} + \\ &\quad s_{t+115} + s_{t+107} + s_{t+85} + s_{t+66} + s_{t+64} + s_{t+52} + s_{t+48} + s_{t+14} + s_t. \end{aligned}$$

A highly nonlinear 17 variable Boolean function of degree 15 is used as the output filtering function. The 17 variables are selected from the states of the LFSR as follows:

$$\begin{aligned} (x_t^{16}, \ldots, x_t^0) \stackrel{def}{=} (&s_{t+212}, s_{t+244}, s_{t+227}, s_{t+193}, s_{t+161}, s_{t+134}, s_{t+105}, s_{t+98}, s_{t+74}, s_{t+58}, \\ &s_{t+44}, s_{t+21}, s_{t+19}, s_{t+9}, s_{t+6}, s_{t+1}, s_t). \end{aligned}$$

In [53] Courtois has studied different kinds of algebraic attacks on SFINKS and presented the vulnerability of this cipher. It is shown that the fast algebraic attack is most efficient than other algebraic attacks discussed in [53].

Attack on E_0 keystream generator

E_0 keystream generator is a part of the the Bluetooth encryption system, used for wireless communication (see, Bluetooth SIG (2001) [13]). The algebraic attacks on E_0 were analysed in [5] and later Armknecht studied implementation of fast algebraic attack [3] which is 8 times faster than the algebraic attack discussed in [5]. This attack can be made faster using the latest improvement on fast algebraic attack in [93].

One may also refer to the papers [52, 1, 2] in this direction.

2.3 Motivation

The above discussed attacks on Toyocrypt, LILI-128, SFINKS and E_0 clearly identify that the attacks are performed exploiting certain kinds of weaknesses in the structure of the underlying Boolean functions. Thus there is a need to carefully study some new cryptographic properties (apart from the existing known cryptographic properties) for any Boolean function when used as a cryptographic primitive.

One should note that the function f or its complement should not have low degree annihilators. That means the Boolean function should have high algebraic immunity. Otherwise

the attacker, knowing some portion of the keystream, can generate low degree equations (see Subsection 2.2.1) which can be solved efficiently to recover the initial key. Further one should note that only having good algebraic degree may not suffice. Given a Boolean function f, having the maximum possible algebraic immunity, should not be used if there exists a low degree function g such that $fg = h$, where the degree of h is the same as the value of maximum algebraic immunity. This is from the view point of fast algebraic attacks.

It may be noted that there are different scenarios when an algebraic attack can be mounted [56, 51]. The properties of the Boolean functions that we consider in this thesis are some necessary conditions to resist certain cases of algebraic and fast algebraic attacks. The actual attacks may also exploit certain other weaknesses in the design rather than only concentrating on the Boolean function. Further some more theory of cryptanalysis may be identified that may force to consider further properties of the Boolean functions to be used in a cryptosystem.

We are motivated by the properties of Boolean functions those are highlighted as necessary conditions to resist certain kinds of algebraic and fast algebraic attacks in recent literature. These properties of the Boolean functions have not been studied yet in a disciplined manner and construction of such functions are explained in details in this thesis. The relationship of these newly found properties with the existing ones of Boolean functions are also studied here. We think the contribution of this thesis will help the cryptosystem designers to choose the functions with more care. From theoretical viewpoint, the Boolean functions are always very elegant and challenging combinatorial objects to study. Our results have also direct implications in development of this subject.

Chapter 3

Study on Algebraic Immunity of Boolean functions

A very well studied model of stream cipher is the nonlinear combiner model, where the outputs of several LFSRs are combined using a nonlinear Boolean function to produce the keystream. This model has undergone a lot of cryptanalysis and to resist those attacks, different design criteria have been proposed for both the LFSRs and the combining Boolean function. The main criteria on the combining function are balancedness, high algebraic degree, high nonlinearity and correlation immunity. Another model is the filter generator, in which the content of some of the flip-flops in a single LFSR constitute the input to a nonlinear Boolean function which produces the keystream. The main criteria on the filtering function are balancedness, high algebraic degree and high nonlinearity. There are large number of important papers in this direction and one may refer to [28, 100, 151, 32] and the references in these papers for more details. Apart from these two models there are number of proposed stream ciphers where Boolean functions are considered as main components and their security depend on the strength of the underlying Boolean functions. Therefore it is necessary to study Boolean functions in terms of their cryptographic properties.

It is known that a Boolean function should be of high algebraic degree to be cryptographically secure [73]. Further, it has been identified recently, that it should not have a low degree multiple [56]. We have already discussed in Subsection 2.2 that given a Boolean function f on n-variables, different kinds of scenarios related to low degree multiples of f have been studied in [56, 123]. It is shown in [56] that given any n-variable Boolean function f, it is always possible to get a Boolean function g with degree at most $\lceil \frac{n}{2} \rceil$ such that fg is of degree at most $\lceil \frac{n}{2} \rceil$. Thus while choosing a Boolean function f, the cryptosystem designer

should be careful that it should not happen that degree of fg falls much below $\lceil \frac{n}{2} \rceil$.

The core of the analysis is to find out minimum (or low) degree annihilators of f and $1+f$, i.e., to find out minimum (or low) degree functions g_1, g_2 such that $fg_1 = 0$ and $(1+f)g_2 = 0$ (see section 2.2.1). To mount algebraic attack, one needs only the low degree linearly independent annihilators [56, 123] of f and $1+f$. At this point, we recapitulate two important issues related to algebraic attack [56, 123].

1. Take $f, g, h \in \mathcal{B}_n$. Assume that there exists a nonzero function g of low degree such that $fg = h$, where h is a nonzero function of low degree and without loss of generality, $\deg(g) \leq \deg(h)$. This is because, if $\deg(g) > \deg(h)$, then $fh = ffg = fg = h$, so one can use h in place of g.

2. Assume there exists a nonzero function g of low degree such that $fg = 0$. This g is called the annihilator of f.

We will now update the notion to consider the multiples of both f and $1+f$.

1. Take $f, g, h \in \mathcal{B}_n$. Assume that there exists a nonzero function g such that $fg = h$ or $(1+f)g = h$, where h is a nonzero function of low degree and without loss of generality, $\deg(g) \leq \deg(h)$. Among all such h's we denote, the lowest degree h (may be more than one and then we take any one of them) by $ldgm_n(f)$.

2. Assume there exists a nonzero function g such that $fg = 0$ or $(1+f)g = 0$. Among all such g's we denote the lowest degree g (may be more than one and then we take any one of them) by $ldga_n(f)$.

For nonzero g and h, $fg = h$ if and only if $(1+f)h = 0$ and $f(g+h) = 0$, i.e., h is an annihilator of $1+f$ and $g+h$ is an annihilator of f and for nonzero g, $fg = 0$ if and only if $(1+f)g = g$. As without loss of generality $\deg(g) \leq \deg(h)$ it can be told that for $f \in \mathcal{B}_n$, $\deg(ldgm_n(f)) = \deg(ldga_n(f))$. Keeping this in mind and returning to the definition of algebraic (annihilator) immunity AI (Definition 11 in Chapter 2) one may note the following. The algebraic (annihilator) immunity of an n-variable Boolean function f is denoted by $\mathsf{AI}_n(f)$ which is basically $\deg(ldgm_n(f))$ or $\deg(ldga_n(f))$. That is the lowest degree of the degree of the functions in $AN(f) \cup AN(1+f)$.

Lemma 1 *[56, 123] Let $f \in \mathcal{B}_n$. Then $\mathsf{AI}_n(f) \leq \lceil \frac{n}{2} \rceil$.*

It is known that there are highly nonlinear Boolean functions of low degree, as example there exist quadratic bent functions (e.g., symmetric bent functions are quadratic), which are of degree 2 and maximum possible nonlinearity $2^{n-1} - 2^{\frac{n}{2}-1}$, when n is even. Such functions f, as they are by themselves of low algebraic degree, will have low values of $\mathsf{AI}_n(f)$. On the other hand, we may have Boolean functions of low nonlinearity with high algebraic degree. Interestingly, this is not the case in terms of AI. In this chapter we show that if a function is of low nonlinearity, then it must have a low value of $\mathsf{AI}_n(f)$. This implies that if one chooses a function with good value of $\mathsf{AI}_n(f)$, that will automatically provide a nonlinearity which is not very low. However it does not assure that the nonlinearity is very high. That is the algebraic immunity property takes care of three fundamental properties of a Boolean function, algebraic degree, weight and nonlinearity, but it does this partially in the case of nonlinearity. Further one may note that this property stays unchanged with respect to affine transformation unlike correlation immunity or propagation characteristics. That is, the algebraic immunities of $f(x)$ and $f(Ax + b)$ are same where $f \in \mathcal{B}_n$, A is a nonsingular $n \times n$ matrix and $b \in \mathbb{F}_2^n$. Here we relate the AI to the Walsh spectra of a Boolean function.

The number of lowest degree linearly independent annihilators are important to generate low degree equations which are exploited to implement algebraic attack. In this context, we also present enumeration results on number of such annihilators.

It is known that a Boolean function must be resilient, should have high nonlinearity and algebraic degree to be used in the nonlinear combiner model of stream cipher. We study such functions for their AI. We present experimental results on highly nonlinear resilient functions which are rotation symmetric [82, 164, 165, 94, 122]. The experiments have been done using [123, Algorithm 1] on functions of $7, 8$ and 9 variables and their complements. The results found are encouraging, which shows that there are highly nonlinear resilient functions which are also optimal in terms of their AI. Further we study different construction methods of resilient functions. We note that the Siegenthaler's construction [161] is not good in terms of AI. On the other hand we show that the construction presented in [136] (basically a construction similar to the Tarannikov's construction [170]) is encouraging in terms of AI. We have also experimentally studied some functions which are of Maiorana-McFarland type [150], i.e., which can be seen as concatenation of affine functions.

Every n-variable Boolean function can be written as concatenation of two $n-1$ variable sub functions (i.e., functions generated by fixing one variable as 0 and 1). There are lots of popular constructions to get n-variable functions by concatenating $n-1$ variable functions. We have studied the AI of a Boolean function in terms of AI of its two sub functions. Then we study how the AI of a Boolean function changes when one add an affine function with it.

3.1 Relationship between AI and Nonlinearity

Let consider $g \in AN(f)$ where $f \in \mathcal{B}_n, n > 0$. As $f * g = 0$, we have $g(x) = 0$ for $x \in \mathbb{F}_2^n$ satisfying $f(x) = 1$. On this context, from [123] we have following result on the relationship among the supports of a Boolean function and its annihilators.

Proposition 1 *[123] Let $f \in \mathcal{B}_n$ and $g \in AN(f)$. Then $supp(f) \subseteq \{x \mid g(x) = 0\}$, i.e., $supp(f) \subseteq supp(1 + g)$.*

Towards proving the results relating AI and the nonlinearity of a Boolean function, we first present the following result where we relate the algebraic degree with the weight of the function.

Theorem 2 *Let $f \in \mathcal{B}_n$ and $\mathsf{AI}_n(f) > d$. Then*

$$\sum_{i=0}^{d} \binom{n}{i} \leq \mathsf{wt}(f) \leq \sum_{i=0}^{n-(d+1)} \binom{n}{i}.$$

Proof : Consider that f has an annihilator g of degree d. Let the ANF of g is

$$a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{1 \leq i < j \leq n} a_{i,j} x_i x_j + \ldots + \sum_{1 \leq i_1 \leq \ldots \leq i_d \leq n} a_{i_1,\ldots,i_d} x_{i_1} \ldots x_{i_d},$$

where a's are from $\mathbb{F}_2$. Note that $f(x) = 1$ implies $g(x) = 0$, since $g \in AN(f)$. So, we will be able to get linear equations from $g(x) = 0$ on the a's in ANF of g when $f(x) = 1$. That is, we will get $\mathsf{wt}(f)$ many homogeneous linear equations on the a's.

Solving the system of homogeneous linear equations, we can find out annihilators g of degree $\leq d$ on nontrivial solutions. In case of a trivial solution we will get all the a's equal to zero, i.e., $g(x) = 0$, which is not acceptable as we are interested in non zero $g(x)$.

Here, we have $\sum_{i=0}^{d} \binom{n}{i}$ number of variables (the a's for the monomials up to degree d) and $\mathsf{wt}(f)$ many number of equations. If the number of variables is greater than the number of equations then we will get nontrivial solutions. Thus f has no annihilator g of degree d implies the number of equations is greater than or equal to the number of variables. So, there must be at least $\sum_{i=0}^{d} \binom{n}{i}$ number of equations, i.e., $\mathsf{wt}(f) \geq \sum_{i=0}^{d} \binom{n}{i}$. Similarly, when considering $1 + f$, we get $\mathsf{wt}(1 + f) \geq \sum_{i=0}^{d} \binom{n}{i}$. This gives, $\mathsf{wt}(f) \leq 2^n - \sum_{i=0}^{d} \binom{n}{i}$, i.e., $\mathsf{wt}(f) \leq \sum_{i=0}^{n-(d+1)} \binom{n}{i}$. ∎

Theorem 2 also gives an alternative proof of $\mathsf{AI}_n(f) \leq \lceil \frac{n}{2} \rceil$ which was given in [123]. The inequality in Theorem 2 will not be satisfied if $d > n - (d+1) \Rightarrow d > \frac{n-1}{2} \Rightarrow d \geq \lceil \frac{n}{2} \rceil$. That is, for any f the inequality in Theorem 2 will not be satisfied if $\mathsf{AI}_n(f) > d \geq \lceil \frac{n}{2} \rceil$.

However, the reverse direction of Theorem 2 is not always true. For example, the affine functions are balanced, i.e., its weight is 2^{n-1}, but clearly they have linear annihilators.

Based on Theorem 2, the following result gives a bound on $\mathsf{wt}(f)$, where f and $1+f$ do not have annihilators of degree less than $\lceil \frac{n}{2} \rceil$.

Corollary 2 $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$ *implies*

1. *f is balanced when n is odd*
2. *$\sum_{i=0}^{\frac{n}{2}-1} \binom{n}{i} \leq \mathsf{wt}(f) \leq \sum_{i=0}^{\frac{n}{2}} \binom{n}{i}$ when n is even.*

Proof : The $\mathsf{wt}(f)$ will satisfy Theorem 2 for $d = \lceil \frac{n}{2} \rceil - 1$. That is

$$\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \binom{n}{i} \leq \mathsf{wt}(f) \leq \sum_{i=0}^{n - \lceil \frac{n}{2} \rceil} \binom{n}{i} \Rightarrow \sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \binom{n}{i} \leq \mathsf{wt}(f) \leq \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{i}.$$

When n is odd, $\lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil - 1$ and hence $\mathsf{wt}(f) = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{i} = 2^{n-1}$. For n even, $\lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{2} \rceil$ and the result follows. ■

Hence, any odd variable unbalanced Boolean function can not have optimal AI.

Proposition 2 *Suppose $f \in \mathcal{B}_{2k}$ for $k \geq 0$ such that f and $1+f$ have no annihilator of degree less than k and $k+1$ respectively. Then*

1. *$\mathsf{wt}(f) = 2^{2k-1} - \binom{2k-1}{k}$.*
2. *$\mathsf{AI}_{2k}(f) = k$.*

Proof : Since f and $1+f$ have no annihilator of degree less than k and $k+1$ respectively, following the proof of the Theorem 2, we have $\mathsf{wt}(f) \geq \sum_{i=0}^{k-1} \binom{n}{i}$ and $\mathsf{wt}(1+f) \geq \sum_{i=0}^{k} \binom{n}{i}$. As $sup(f) \cup sup(1+f) = \mathbb{F}_2^{2k}$ and $|\mathbb{F}_2^{2k}| = 2^{2k}$, $\mathsf{wt}(f) = \sum_{i=0}^{k-1} \binom{n}{i}$ and $\mathsf{wt}(1+f) = \sum_{i=0}^{k} \binom{n}{i}$.

These weight constraints imply that f and $1+f$ have annihilators of degree k and $k+1$ respectively. Hence, $\mathsf{AI}_{2k}(f) = k$ and $\mathsf{wt}(f) = 2^{2k-1} - \binom{2k-1}{k}$. ■

Now we connect AI with nonlinearity.

Theorem 3 *If* $\mathsf{nl}(f) < \sum_{i=0}^{d} \binom{n}{i}$, *then* $\mathsf{AI}_n(f) \leq d+1$.

Proof : Let $\alpha \in \mathbb{F}_2^n$ such that $|W_f(\alpha)|$ is maximum, i.e., $\mathsf{nl}(f) = \min_{\alpha \in \mathbb{F}_2^n} \{\mathsf{wt}(f + \alpha \cdot x), \mathsf{wt}(1 + f + \alpha \cdot x)\}$. We use the contrapositive result of Theorem 2. If d is the least integer such that $\min_{\alpha \in \mathbb{F}_2^n} \{\mathsf{wt}(f + \alpha \cdot x), \mathsf{wt}(1 + f + \alpha \cdot x)\} < \sum_{i=0}^{d} \binom{n}{i}$ then $\mathsf{AI}_n(f + \alpha \cdot x) \leq d$. Now following the Proposition 4 (see later) we have $\mathsf{AI}_n(f) \leq d+1$ as $\alpha \cdot x$ is an affine function on the variables. ■

From the above theorem we directly get the following result.

Corollary 3 *If* $\mathsf{AI}_n(f) > d+1$ *then* $\mathsf{nl}(f) \geq \sum_{i=0}^{d} \binom{n}{i}$, *i.e.,* $\mathsf{nl}(f) \geq \sum_{i=0}^{\mathsf{AI}_n(f)-2} \binom{n}{i}$.

In the proof we exploited the Walsh spectrum value only at zero vector (i.e., the weight of function f). Motivated by our idea, later Lobanov [111] studied the vector (say v) where the absolute value of Walsh spectrum is maximum and then the weight of the function $f + \langle v, x \rangle$ to present the strict lower bound on the nonlinearity of f. The result is as follows:

Theorem 4 *[111] Let* $f \in \mathcal{B}_n$ *and* $\mathsf{AI}_n(f) = k$. *Then* $\mathsf{nl}(f) \geq 2^{n-1} - \sum_{i=k-1}^{n-k} \binom{n-1}{i} = 2\sum_{i=0}^{k-2} \binom{n-1}{i}$.

This bound improves upon the corresponding bound of Theorem 3 and is strict. We will present examples of Boolean functions having optimal AI in Chapters 4 and 5 which attain this lower bound. In the following theorem, it has been further generalized by Carlet in [35] to a bound on higher order nonlinearity. On this context, we define for $r > 0$, the r-th order nonlinearity of an n variable Boolean function f as the minimum distance of f from all the functions of algebraic degree at most r, i.e., $\mathsf{nl}_r(f) = \min_{\{h \mid \deg(h) \leq r\}} d(f, h)$. Here 1st order nonlinearity (for $r = 1$) is the nonlinearity, we defined in Definition 6.

Theorem 5 *[35] Let $f \in \mathcal{B}_n$ and r be a positive integer. The nonlinearity of order r satisfies* $\mathsf{nl}_r(f) \geq 2 \sum_{i=0}^{\mathsf{AI}_n(f)-r-1} \binom{n-r}{i}$.

The interesting situation is when $\deg(f) > d+1$. Because, when $\deg(f) \leq d+1$, then irrespective of the nonlinearity of f, $\mathsf{AI}_n(f) \leq d+1$, since $f*(1+f)=0$. As there are low degree functions with very high nonlinearity (as example quadratic bent function), it is clear that there are functions f with high nonlinearity and low $\mathsf{AI}_n(f)$ (basically $1+f$). Then Theorem 3 and the result of [111] give a new reason why one should not use functions f with low nonlinearity, since in that case $\mathsf{AI}_n(f)$ would be low. A function with $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$, by itself, takes care of good nonlinearity and algebraic degree. However, they do not assure that if f has high algebraic immunity (for instance an optimum one $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$) then its nonlinearity will be very high. Indeed, the result of [111] implies then that f has nonlinearity at least $2\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 2} \binom{n-1}{i}$, that is, $2^{n-1} - \binom{n-1}{\frac{n-1}{2}}$ if n is odd and $2^{n-1} - \binom{n-1}{\frac{n}{2}-1} - \binom{n-1}{\frac{n}{2}}$ if n is even.

3.2 Count of Annihilators

In the proof of Theorem 2, we get $\mathsf{wt}(f)$ many homogeneous linear equations using the a's. Let us denote the coefficient matrix of this system of equations by M. Then M has $\mathsf{wt}(f)$ many rows and $\sum_{i=0}^{d} \binom{n}{i}$ many columns. The rank (say, r) of the matrix M, $r \leq \min\{\mathsf{wt}(f), \sum_{i=0}^{d} \binom{n}{i}\}$.

1. If $r = \sum_{i=0}^{d} \binom{n}{i}$, then there is no annihilator of degree $\leq d$.
2. If $r < \sum_{i=0}^{d} \binom{n}{i}$, then there are annihilators of degree $\leq d$. There will be $\sum_{i=0}^{d} \binom{n}{i} - r$ many linearly independent annihilators having degree $\leq d$.

For any Boolean function f, the number of annihilators and linearly independent annihilators are $2^{\mathsf{wt}(1+f)}-1$ and $\mathsf{wt}(1+f)$ respectively. It is clear [123] that a larger number of low degree linearly independent annihilators helps better in one as cryptanalyser can generate larger number of low degree equations. Thus when considering a Boolean function one should check the number of independent annihilators at the lowest possible degree. Towards this we present some enumeration results on lowest degree annihilators.

Definition 12 *Given $f \in \mathcal{B}_n$, by $\#LDA_n(f)$, we denote the number of independent annihilators of f at $\mathsf{AI}_n(f)$ i.e., $\#LDA_n(f) = |AN_{\mathsf{AI}_n(f)}(f)|$.*

Theorem 6

1. *Take $f \in \mathcal{B}_n$, with $\mathsf{AI}_n(f) = d+1 < \lceil \frac{n}{2} \rceil$. Then $\#LDA_n(f) \leq \binom{n}{d+1}$.*
2. *Take balanced $f \in \mathcal{B}_n$, n even with $\mathsf{AI}_n(f) = \frac{n}{2}$. Then $\#LDA_n(f) \geq \frac{\binom{n}{\frac{n}{2}}}{2}$.*
3. *Take a balanced function $f \in \mathcal{B}_n$, n odd such that $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$. Then $\#LDA_n(f) = \binom{n}{\lceil \frac{n}{2} \rceil}$.*

Proof : The proof of item 1 is as follows. It is given that there is no annihilator up to degree d. If one considers an annihilator of degree d, then the only solution would become the trivial zero function. The rank of the coefficient matrix M is equal to number of variables, i.e., equal to $\sum_{i=0}^{d} \binom{n}{i}$. Now the function f has annihilator at degree $d+1$. The corresponding coefficient matrix (say M') is obtained from M by adding $\binom{n}{d+1}$ columns. Thus the rank of M' will be greater or equal to the rank of M, i.e., $\sum_{i=0}^{d} \binom{n}{i}$. Number of independent solutions will be less than or equal to $\sum_{i=0}^{d+1} \binom{n}{i} - \sum_{i=0}^{d} \binom{n}{i} = \binom{n}{d+1}$.

Now we prove item 2. Here $\mathsf{wt}(f) = 2^{n-1}$. The function f has annihilator at degree $\frac{n}{2}$. In this case the corresponding coefficient matrix (say M) will have 2^{n-1} many rows and $\sum_{i=0}^{\frac{n}{2}} \binom{n}{i} = 2^{n-1} + \frac{\binom{n}{\frac{n}{2}}}{2}$ many columns. Thus rank of M will be $\leq 2^{n-1}$. Number of independent solutions will be greater than or equal to $(2^{n-1} + \frac{\binom{n}{\frac{n}{2}}}{2}) - 2^{n-1} = \frac{\binom{n}{\frac{n}{2}}}{2}$.

Here we present the proof of item 3. Here $\mathsf{wt}(f) = 2^{n-1}$. It is given that there is no non zero annihilator up to degree $\lfloor \frac{n}{2} \rfloor$. If one considers an annihilator of degree $\leq \lfloor \frac{n}{2} \rfloor$, then the only solution would become the trivial zero function. In this case the number of variables (the a's) is $\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{i} = 2^{n-1}$. So the coefficient matrix M is a $2^{n-1} \times 2^{n-1}$ square matrix. As it has no nontrivial solution, its rank $r = 2^{n-1}$. The function f has annihilator at degree $\lceil \frac{n}{2} \rceil$. In this case the corresponding coefficient matrix (say M') will have 2^{n-1} many rows and $2^{n-1} + \binom{n}{\lceil \frac{n}{2} \rceil}$ many columns. Thus rank of M' will be equal to that of M, i.e., 2^{n-1}. The number of independent solutions equals to $(2^{n-1} + \binom{n}{\lceil \frac{n}{2} \rceil}) - 2^{n-1}$, i.e., $\binom{n}{\lceil \frac{n}{2} \rceil}$. ∎

In the next section, we will study the algebraic immunity of Boolean functions in terms of algebraic immunity of its sub functions.

3.3 AI of a Boolean Function in terms of the AI of its Sub functions

In the following proposition we present the AI of a Boolean function in terms of the AI of its sub functions after fixing one variable. For notational purpose, given $f \in \mathcal{B}_n$, we denote the set $LDGA_n(f)$ as the set of lowest degree f_1's ($f_1 \in \mathcal{B}_n$) such that $f*f_1 = 0$ or $(1+f)*f_1 = 0$, i.e., $LDGA_n(f) = AN_{\mathsf{AI}_n(f)}(f) \cup AN_{\mathsf{AI}_n(f)}(1+f)$.

Proposition 3 *Let $f, g \in \mathcal{B}_n$ on variables $x_1, x_2, \cdots, x_n$ with $\mathsf{AI}_n(f) = d_1$ and $\mathsf{AI}_n(g) = d_2$. Let $h = (1+x_{n+1})f + x_{n+1}g \in \mathcal{B}_{n+1}$. Then*

1. *if $d_1 \neq d_2$ then $\mathsf{AI}_{n+1}(h) = \min\{d_1, d_2\} + 1$.*
2. *Given $d_1 = d_2 = d$, $d \leq \mathsf{AI}_{n+1}(h) \leq d+1$. Further, $\mathsf{AI}_{n+1}(h) = d$ iff there exists $f_1, g_1 \in \mathcal{B}_n$ of algebraic degree d such that $\{f*f_1 = 0, g*g_1 = 0\}$ or $\{(1+f)*f_1 = 0, (1+g)*g_1 = 0\}$ and $\deg(f_1+g_1) \leq d-1$.*

Proof : Let $f_1 \in LDGA_n(f)$ and $g_1 \in LDGA_n(g)$. Thus, either $f*f_1 = 0$ which gives $(1+x_{n+1})*f_1*h = 0$ or $(1+f)*f_1 = 0$ which gives $(1+x_{n+1})*f_1*(1+h) = 0$. Also either $g*g_1 = 0$ implies $x_{n+1}*g_1*h = 0$ or $(1+g)*g_1 = 0$ implies $x_{n+1}*g_1*(1+h) = 0$. Thus,

$$\mathsf{AI}_{n+1}(h) \leq \min\{\mathsf{AI}_n(f), \mathsf{AI}_n(g)\} + 1. \tag{3.1}$$

Let $p = (1+x_{n+1})p_1 + x_{n+1}p_2 \in LDGA_{n+1}(h)$ where at least one of the p_1 and p_2 is nonzero. Let us first consider the case with $h*p = 0$ which implies $(1+x_{n+1})f*p_1 + x_{n+1}g*p_2 = 0$. So $f*p_1 = 0$ and $g*p_2 = 0$. Similarly for the case with $(1+h)*p = 0$, i.e., $(1+x_{n+1})*(1+f)*p_1 + x_{n+1}(1+g)*p_2 = 0$, we have $(1+f)*p_1 = 0$ and $(1+g)*p_2 = 0$. Now there could be three cases in both the scenarios.

(a) p_1 is zero, but p_2 is non zero. So $\deg(p_2) \geq d_2$ which gives $\deg(p) \geq d_2 + 1$.

(b) p_2 is zero, but p_1 is non zero. So $\deg(p_1) \geq d_1$ which gives $\deg(p) \geq d_1 + 1$.

(c) Both p_1, p_2 are non zero. So $\deg(p_1) \geq d_1$ and $\deg(p_2) \geq d_2$, which gives $\deg(p) \geq \max\{d_1, d_2\} + 1$, when $d_1 \neq d_2$.

So, from these three scenarios, for $d_1 \neq d_2$ we get,

$$\mathsf{AI}_{n+1}(h) \geq \min\{\mathsf{AI}_n(f), \mathsf{AI}_n(g)\} + 1. \tag{3.2}$$

Equation 3.1, 3.2 give the proof of item 1.

Now we prove item 2. Consider $p = (1 + x_{n+1})f_1 + x_{n+1}g_1 \in LDGA_{n+1}(h)$. So, $\deg(p) \geq \deg(f_1) = d_1$. It could happen that all highest degree terms of $x_{n+1}f_1 + x_{n+1}g_1$ in p get canceled and the over all degree is decreased by one. So, $d \leq \mathsf{AI}_{n+1}(h) \leq d + 1$.

Let $\mathsf{AI}_{n+1}(h) = d$. Then the highest degree terms of f_1 and g_1 must be same which gives $deg(f_1 + g_1) \leq d - 1$. Now we prove the other side. Let there exist $f_1, g_1 \in \mathcal{B}_n$ of degree d such that $\deg(f_1 + g_1) \leq d - 1$ and one of the following holds

$$f * f_1 = 0, g * g_1 = 0, \tag{3.3}$$

$$(1 + f) * f_1 = 0, (1 + g) * g_1 = 0. \tag{3.4}$$

Construct $p = (1 + x_{n+1})f_1 + x_{n+1}g_1$. Thus $h * p = 0$ (when Equation 3.3 is considered) or $(1 + h) * p = 0$ (when Equation 3.4 is considered). So, $\mathsf{AI}_{n+1}(h) = d$. ■

As a result, if AI of one of the sub functions (fixing one variable) is bad, then the AI of the function will be bad. The next corollary is a direct consequence of Proposition 3 and of the upper bound $\lceil \frac{n}{2} \rceil$ on the algebraic immunity of n-variable functions.

Corollary 4 *Let $h = (1 + x_{n+1})f + x_{n+1}g \in \mathcal{B}_{n+1}$ where n is even and $\mathsf{AI}_{n+1}(h) = \frac{n}{2} + 1$ (i.e., has maximum possible value). Then $\mathsf{AI}_n(f) = \mathsf{AI}_n(g) = \frac{n}{2}$ (i.e., is maximum) and there do not exist $f_1, g_1 \in \mathcal{B}_n$ of degree $\frac{n}{2}$ such that "$f * f_1 = 0$ and $g * g_1 = 0$" or "$(1 + f) * f_1 = 0$ and $(1 + g) * g_1 = 0$" and such that all $\frac{n}{2}$ degree monomials of f_1 and g_1 are same.*

In the following corollary we observe that two functions on an odd number of variables with optimum algebraic immunity always have some relationship.

Corollary 5 *Let $f, g \in \mathcal{B}_n$ where n is odd and $\mathsf{AI}_n(f) = \mathsf{AI}_n(g) = \frac{n+1}{2}$ (the maximum possible value). Then there must exist $f_1, g_1 \in \mathcal{B}_n$ of degree $\frac{n+1}{2}$ such that "$f * f_1 = 0$ and $g * g_1 = 0$" or "$(1 + f) * f_1 = 0$ and $(1 + g) * g_1 = 0$" and such that all $\frac{n+1}{2}$ degree monomials of f_1 and g_1 are same.*

Proof : Let $h = (1 + x_{n+1})f + x_{n+1}g \in \mathcal{B}_{n+1}$. According to Proposition 3, $\mathsf{AI}_{n+1}(h)$ equals $\frac{n+1}{2}$ since it cannot be greater than $\frac{n+1}{2}$. Again according to Item 2 of Proposition 3 we have the proof. ■

Corollary 6 *Let $f \in \mathcal{B}_n$, $\mathsf{AI}_n(f) = d$ and $h = x_{n+1} + f \in \mathcal{B}_{n+1}$.*

1. *Then* $d \leq \mathsf{AI}_{n+1}(h) \leq d+1$.

2. $\mathsf{AI}_n(h) = d$ *iff there exist* $f_1, f_2 \in LDGA_n(f)$ *such that* $f * f_1 = 0$, $(1+f) * f_2 = 0$ *and* $\deg(f_1 + f_2) \leq d-1$.

Proof : Since $x_{n+1} + f = (1 + x_{n+1})f + x_{n+1}(1 + f)$, this follows from Proposition 3. ∎

In the same line we present one more technical result.

Proposition 4 *Let* $f(x_1, \ldots, x_n) \in \mathcal{B}_n$ *and* $\mathsf{AI}_n(f) = d$. *Let* l *be a affine function with any of the following properties: (i)* l *is a function on* $x_1, \ldots, x_n$, *(ii)* l *is a function on variables other than* $x_1, \ldots, x_n$, *(iii)* l *is a function on* $x_1, \ldots, x_n$ *and some other variables. Let* $l + f$ *be a function on* m *variables. Then* $d - 1 \leq \mathsf{AI}_m(l + f) \leq d + 1$ *for cases (i) and (iii) and* $d \leq \mathsf{AI}_m(l + f) \leq d + 1$ *for case (ii).*

Proof : Let $g \in LDGA_n(f)$, which implies $f*g = 0$ or $(1+f)*g = 0$ and $\deg(g) = d$. So for any affine function l, we have $(l+f)*((1+l)*g) = 0$ if $f*g = 0$ or $(l+f+1)*((1+l)*g) = 0$ if $(1 + f) * g = 0$. Hence, $\mathsf{AI}_m(f + l) \leq d + 1$. So, the upper bound for all cases is proved.

Now we consider case (i), where l is an affine function on the variables $x_1, \ldots, x_n$. Let there be an $l \in A_n$ such that $\mathsf{AI}_n(f + l) < d - 1$. Then $\mathsf{AI}_n(f) = \mathsf{AI}_n((f + l) + l) \leq \mathsf{AI}_n(f + l) + 1 < d$, which contradicts that $\mathsf{AI}_n(f) = d$. Thus, $\mathsf{AI}_m(l + f) \geq d - 1$.

The lower bound of case (ii) follows from repeated application of Corollary 6.

Now we prove lower bound of case (iii). Let $l = l_1 + l_2$, where l_1 is an affine function on some or all of the variables $x_1, \ldots, x_n$ and l_2 is an affine function on some other variables. So, following case (i), we have $\mathsf{AI}_n(f + l_1) \geq d - 1$. Then following case (ii), $\mathsf{AI}_m(f + l) = \mathsf{AI}_m((f + l_1) + l_2) \geq \mathsf{AI}_m(f + l_1) = d - 1$. ∎

3.3.1 Functions with Low Degree sub functions

In this sub section we discuss why a Boolean function with low degree sub function is not good in terms of algebraic immunity. This result is a generalization of the result presented in [123], where the authors have shown that certain kind of Maiorana-McFarland constructions are not good in terms of algebraic immunity.

Proposition 5 *Let* $f \in \mathcal{B}_n$. *Let* $g \in \mathcal{B}_{n-r}$ *be a sub function of* $f(x_1, \ldots, x_n)$ *after fixing* r *many distinct inputs* $x_{i_1}, \ldots, x_{i_r} \in \{x_1, \ldots, x_n\}$. *If the algebraic degree of* g *is* d, *then* $\mathsf{AI}_n(f) \leq d + r$.

Proof : Let $x_{i_1}, \ldots, x_{i_r}$ are fixed at the values $a_{i_1}, \ldots, a_{i_r} \in \mathbb{F}_2$. Thus g is a function on the variables $\{x_1, \ldots, x_n\} \setminus \{x_{i_1}, \ldots, x_{i_r}\}$. It can be checked that $(1 + a_{i_1} + x_{i_1}) \ldots (1 + a_{i_r} + x_{i_r})(1+g)$ is an annihilator of f. The algebraic degree of $(1+a_{i_1}+x_{i_1}) \ldots (1+a_{i_r}+x_{i_r})(1+g)$ is $d + r$. Thus the result. ■

The Maiorana-McFarland construction can be seen as concatenation of affine functions on $n - r$ variables to construct an n-variable functions. Clearly we have affine sub functions of the constructed function in this case and hence $\deg(g) = 1$ following the notation of Proposition 5. Thus there will be annihilators of degree $1 + r$. Note that if r is small, then one can get annihilators at low degree [123, Theorem 2, Example 1]. This situation for Maiorana-McFarland construction is only a sub case of our proposition. Our result works on any function, it need not be of Maiorana-McFarland type only. We present an example below.

Example 4 *Let us consider a* 20*-variable function, with a sub function of degree* 2 *on* 17*-variables, i.e., we fix* 3 *inputs. In that case the* 20*-variable function will have an annihilator at degree* $2 + 3 = 5$*.*

It should be noted that the converse of Proposition 5 is not always true. That is, a function having low degree annihilator does not imply it always has some low degree sub function by fixing a few variables. As example, one may refer to the 5-variable function $f = x_1 + x_2 + x_2x_4 + x_3x_4 + (x_2 + x_3 + x_1x_4 + x_2x_4 + x_3x_4)x_5$. This function has algebraic immunity 2 and the only annihilator of degree 2 is $1+x_1+x_2+x_1x_4+x_3x_4+(x_2+x_3+x_4)x_5$. If one verifies all possible sub functions of f after fixing 1 and 2 variables, it is not possible to get sub functions of degree 1 and 0 respectively.

It will be interesting to extend our idea on the Boolean functions that can be seen as concatenation of indicators of flats [34].

In the next section, we will study certain constructions of cryptographically significant Boolean functions in terms of algebraic immunity.

3.4 Studying Existing Functions for Their AI

It has been in [123, 68] that any randomly chosen balanced function on large number of variables will have good algebraic immunity with very high probability. This result is in a similar direction that most of the Boolean functions are of high algebraic degree or of high

nonlinearity in general. That is if one chooses a Boolean function randomly, the probability that these properties will be good is high. However, when considering a specific construction technique, the number of functions constructed by that method is much lower than the total space of Boolean functions and generally such statistical analysis does not work.

3.4.1 Experimental Results on Rotation Symmetric Boolean Functions

Let us consider that we want to find (n, m, d, x) functions (n-variable, m-resilient, degree d and nonlinearity x) with best possible parameters along with the best possible algebraic immunity. In this direction we first refer to a small subset of Boolean functions, the rotation symmetric Boolean functions (RSBFs)(See Section 2.1.5 of Chapter 2 for definition). We present experimental results related to the algebraic immunity of the RSBFs which are available in [164, 165, 94, 122].

Experiment 1: First we test the algebraic immunity for $(7, 2, 4, 56)$ RSBFs. It is given in [164] that there are 36 such functions with $f(0) = 0$. Out of them, 24 functions contain linear terms. For these functions, $\mathsf{AI}_n(f) = 3$, which is 1 less than highest value $\lceil \frac{n}{2} \rceil = 4$. Out of them 12 functions have $\#LDA_n(f) = 3$ and the rest 12 have $\#LDA_n(f) = 4$. The algebraic immunity of the other 12 functions, where the linear terms are not there, $\mathsf{AI}_n(f) = 4$, which is the highest possible value. According to Theorem 6(item 3) (we have also checked by experiment), for these functions $\#LDA_n(f) = \binom{7}{\lceil \frac{7}{2} \rceil} = 35$.

Experiment 2: Here we examine the $(8, 1, 6, 116)$ RSBFs with $f(0) = 0$ which are 10272 in number [165]. Out of them, 6976 numbers attains highest algebraic immunity, i.e., 4 and we find that for these functions $\#LDA_n(f) = 35$. From Theorem 6(item 2), in this case the value should be $\geq \frac{\binom{8}{\frac{8}{2}}}{2} = 35$. Thus we find an example, where the bound is tight. For the rest $10272 - 6976 = 3296$ functions, the algebraic immunity is 3. Out of them 1536 many functions f have only one annihilator at degree 3 (but no degree 3 annihilator for $1+f$), 1504 many functions f have no annihilator at degree 3 (but one degree 3 annihilator for $1 + f$) and 256 many functions f have one annihilator at degree 3 (also one degree 3 annihilator for $1 + f$). According to Theorem 6(item 1), $\#LDA_n(f) \leq \binom{8}{3} = 56$. So for these functions, the bound is not sharp.

Experiment 3: In the above two experiments, we examined the functions which are balanced. Now we consider the $[9, 3, 5, 240]$ RSBFs which are not balanced. We consider the functions with $f(0) = 0$, and these are 8406 in number [94, 122]. According to Corol-

lary 2(item 1), the algebraic immunity of these functions will be strictly less than 5. Here after experiment we get the algebraic immunity of all 8406 functions as 4. From Theorem 6(item 1), $\#LDA_9(f) \leq \binom{9}{4} = 126$. In Table 3.4.1, we present the number of functions satisfying a particular $\#LDA_9(f)$ and $\#LDA_9(1+f)$.

$\#LDA_9(f)$	16	17	18	19	20	21
$\#LDA_9(1+f)$	0	1	2	3	4	5
$\#f$	5658	1758	774	180	12	24

Studying the resilient functions on 7 and 8 variables and unbalanced correlation immune functions on 9-variables for this rotation symmetric class of Boolean functions, it is evident that there exists functions which are good in terms of algebraic immunity. It will be interesting to study such functions on higher number of variables.

3.4.2 Analysis of Some Construction Methods

In this section we study some popular constructions in terms of algebraic immunity.

Siegenthaler Construction

In [161] Siegenthaler proposed a construction of resilient functions. Take an initial (n, m, d, σ) function $f(x_1, \ldots, x_n)$. The function $F(x_1, \ldots, x_{n+k}) = x_{n+k} + \ldots + x_{n+1} + f(x_1, \ldots, x_n)$ will be an $(n+k, m+k, d, 2^k\sigma)$ one. From Proposition 4, we get $\mathsf{AI}_n(f) \leq \mathsf{AI}_{n+k}(F) \leq \mathsf{AI}_n(f)+1$. Thus this construction is not good in terms of algebraic immunity.

Modified Tarannikov Construction in [136]

In [170], Tarannikov has proposed an important construction of resilient functions and based on that a similar kind of construction has been proposed in [136]. We will refer the construction in [136] here and study the algebraic immunity of such functions. Let us first present the construction.

An $(n, m, d, -)$ function f is called to be in *desired* form if it is of the form $f = (1 + x_n)f_1 + x_n f_2$, where f_1, f_2 are $(n-1, m, d-1, -)$ functions. Let f be an (n, m, d, σ) function in *desired* form, where f_1, f_2 are both $(n-1, m, d-1, -)$ functions. Let

$$F = x_{n+2} + x_{n+1} + f \text{ and}$$

$$G = (1 + x_{n+2} + x_{n+1})f_1 + (x_{n+2} + x_{n+1})f_2 + x_{n+2} + x_n.$$

In the language of [170], the function G above is said to depend quasilinearly on the pair of variables (x_{n+2}, x_{n+1}). We construct a function H in $n+3$ variables in the following way,

$$H = (1 + x_{n+3})F + x_{n+3}G.$$

Then the function H constructed from f is an $(n+3, m+2, d+1, 2^{n+1} + 4\sigma)$ function in the *desired* form. Thus, this construction can be applied iteratively.

Construction 1 *Let us describe this construction with some index to present the iterative effect. Let H^0 be the initial function of n variables and H^i be the constructed function after i-th iteration. Denote $H^{i'}$ as the function generated from H^i by replacing the variable x_{n+3i} by $(x_{n+3i+2} + x_{n+3i+1})$. Let $F^{i+1} = x_{n+3i+2} + x_{n+3i+1} + H^i$ and $G^{i+1} = x_{n+3i+2} + x_{n+3i} + H^{i'}$. Then the constructed function at $i+1$-th step, $H^{i+1} = (1 + x_{n+3i+3})F^{i+1} + x_{n+3i+3}G^{i+1}$.*

Now we present a technical result.

Proposition 6 *For $i > 0$, $H^i = (1 + Y_i)H^0 + Y_iH^{0'} + Z_i$ where $\deg(Y_i) = i$ and $\deg(Z_i) = i + 1$.*

Proof : The base case is as follows.

$$\begin{aligned} H^1 &= (1 + x_{n+3})F^1 + x_{n+3}G^1 \\ &= (1 + x_{n+3})H^0 + x_{n+3}H^{0'} + (1 + x_{n+3})(x_{n+2} + x_{n+1}) + x_{n+3}(x_{n+2} + x_n) \\ &= (1 + Y_1)H^0 + Y_1H^{0'} + Z_1, \end{aligned}$$

where Y_1 is a 1-degree polynomial and Z_1 is a 2-degree polynomial.

Let us assume that this is true for some $k \geq 1$, i.e., $H^k = (1+Y_k)H^0 + Y_kH^{0'} + Z_k$, where Y_k is a k-degree polynomial and Z_k is $k+1$-degree polynomial. Now,

$$\begin{aligned} H^{k+1} &= (1 + x_{n+3k+3})(x_{n+3k+2} + x_{n+3k+1} + H^k) \\ &\quad + x_{n+3k+3}(x_{n+3k+2} + x_{n+3k} + H^{k'}) \\ &= (1 + x_{n+3k+3})H^k + x_{n+3k+3}H^{k'} \\ &\quad + (1 + x_{n+3k+3})(x_{n+3k+2} + x_{n+3k+1}) + x_{n+3k+3}(x_{n+3k+2} + x_{n+3k}) \\ &= (1 + x_{n+3k+3})((1 + Y_k)H^0 + Y_kH^{0'} + Z_k) \\ &\quad + x_{n+3k+3}((1 + Y_k')H^0 + Y_k'H^{0'} + Z_k') \\ &\quad + (1 + x_{n+3k+3})(x_{n+3k+2} + x_{n+3k+1}) + x_{n+3k+3}(x_{n+3k+2} + x_{n+3k}), \end{aligned}$$

where ${Y_k}'$ and ${Z_k}'$ are generated by replacing the variable x_{n+3k} by $(x_{n+3k+2} + x_{n+3k+1})$ in Y_k and Z_k respectively. Thus,

$$\begin{aligned} H^{k+1} &= (1 + Y_k + Y_k x_{n+3k+3} + {Y_k}' x_{n+3k+3}) H^0 \\ &+ (Y_k + Y_k x_{n+3k+3} + {Y_k}' x_{n+3k+3}) H^{0'} + (1 + x_{n+3k+3}) Z_k + x_{n+3k+3} {Z_k}' \\ &+ (1 + x_{n+3k+3})(x_{n+3k+2} + x_{n+3k+1}) + x_{n+3k+3}(x_{n+3k+2} + x_{n+3k}). \end{aligned}$$

This implies, $H^{k+1} = (1 + Y_{k+1})H^0 + Y_{k+1}H^{0'} + Z_{k+1}$, where Y_{k+1} and Z_{k+1} are $k+1$ and $k+2$ degree polynomials respectively. ■

Now we present the lower and upper bound on algebraic immunity of H^i in terms of the algebraic immunity of H^0.

Theorem 7 $\mathsf{AI}_n(H_0) \leq \mathsf{AI}_{n+3i}(H_i) \leq \mathsf{AI}_n(H_0) + i + 2$.

Proof : To show $\mathsf{AI}_n(H_0) \leq \mathsf{AI}_{n+3i}(H_i)$, it is enough to show $\mathsf{AI}_{n+3}(H^1) \geq \mathsf{AI}_n(H^0)$. We have $H^1 = (1 + x_{n+3}) * F^1 + x_{n+3} * G^1$ where $F^1 = x_{n+2} + x_{n+1} + H^0$ and $G^1 = x_{n+2} + x_n + H^{0'}$. Let $\mathsf{AI}_n(H^0) = d$. So, $\mathsf{AI}_n(H^{0'}) = d$. Following Proposition 4[case (ii)] we have $\mathsf{AI}_{n+2}(F^1) \geq d$, and following Proposition 4[case (iii)] we have $\mathsf{AI}_{n+2}(G^1) \geq d - 1$. Then following Proposition 3, we have $\mathsf{AI}_n(H^1) \geq d$.

Now we prove the upper bound. Following Proposition 6, we get $H^i = (1+Y_i)H^0 + Y_i H^{0'} + Z_i$, where Y_i and Z_i are degree i and degree $i+1$ polynomials respectively. Let algebraic immunity of H^0 be d. Let there be a polynomial g^0 having degree d such that $H^0 * g^0 = 0$ or $(1 + H^0) * g^0 = 0$. Let $H^0 = p + q * x_n$ where p, q are functions on $n-1$ variables, free from the variable x_n. So, $(1+Y_i)H^0 + Y_i H^{0'} = (1+Y_i) * (p + q * x_n) + Y_i * (p + q * (x_{n+1} + x_{n+2})) = Y_i * q * (x_n + x_{n+1} + x_{n+2}) + p + q * x_n = Y_i * q * (x_n + x_{n+1} + x_{n+2}) + H^0$.

Construct a function $U = g^0 * (1 + Z_i) * (1 + x_n + x_{n+1} + x_{n+2})$ of degree at most $d + i + 2$. Now, if $H^0 * g^0 = 0$ then $H^i * U = ((1 + Y_i)H^0 + Y_i H^{0'} + Z_i) * U = (Y_i * q * (x_n + x_{n+1} + x_{n+2}) + H^0 + Z_i) * g^0 * (1 + Z_i) * (1 + x_n + x_{n+1} + x_{n+2}) = 0$. Similarly for $(1 + H^0) * g^0 = 0$, it can be shown that $(1 + H^i) * U = 0$. ■

During each iteration, the algebraic immunity increases at most by 2. This is because, $H^{i+1} = (1 + x_{n+3i+3})(H^i + x_{n+3i+2} + x_{n+3i+1}) + x_{n+3i+3}(H^{i'} + x_{n+3i+2} + x_{n+3i})$. If $g, h \in \mathcal{B}_n$ and $\deg(g), \deg(h) \leq d$ such that $H^i * g = h$ then $H^{i+1} * (1 + x_{n+3i+3}) * g = (1 + x_{n+3i+3})(h + g * (x_{n+3i+2} + x_{n+3i+1}))$, which shows algebraic immunity can not increase by more than two during each iteration. On the other hand, if we go for i many iterations, then the maximum increase in algebraic immunity is $i + 2$.

Later in [18], the authors have proved that the algebraic immunity of the n-variable functions constructed by Construction 1 attain $\mathbf{\Omega}(\sqrt{n})$ algebraic immunity. Theoretically, this [18] presents a sharper result than our result in terms of analysing Tarannikov's construction [170, 136].

Example 5 *Let us start with an initial* $(5, 1, 3, 12)$ *function* $H^0 = x_5(x_1x_4 + x_3x_4 + x_2x_4 + x_2 + x_3) + x_1x_4 + x_3x_4 + x_2 + x_1$. *We found the algebraic immunity of* H^0, H^1, H^2, H^3 *are* $2, 4, 4, 5$ *respectively. The function* H^1 *is an* $(8, 3, 4, 112)$ *function with* $\mathsf{AI}_8(H^1) = 4$. *This function is optimized considering order of resiliency, nonlinearity, algebraic degree and algebraic immunity together. The function* H^2 *is an* $(11, 5, 5, 992)$ *function. Since the algebraic degree of this function is* 5, *we cannot have* $\mathsf{AI}_{11}(H^2)$ *as high as* $\lceil \frac{11}{2} \rceil = 6$, *we can get the value* 5 *at maximum. We checked that the value is actually* $\mathsf{AI}_{11}(H^2) = 4$. *The function* H^3 *is a* $(14, 7, 6, 2^{13} - 2^8)$ *function. Since the algebraic degree of this function is* 6, *we cannot have* $\mathsf{AI}_{14}(H^3)$ *as high as* $\frac{14}{2} = 7$, *we can get the value* 6 *at maximum. We checked that the value is actually* $\mathsf{AI}_{14}(H^3) = 5$.

The Maiorana-McFarland Construction

The original Maiorana-McFarland class of bent function is as follows (see e.g. [27]). Consider n-variable Boolean functions on (x, y), where $x, y \in \mathbb{F}_2^{\frac{n}{2}}$ of the form $f(x, y) = x \cdot \pi(y) + g(y)$ where π is a permutation on $\mathbb{F}_2^{\frac{n}{2}}$ and g is any Boolean function on $\frac{n}{2}$ variables. The function f can be seen as concatenation of $2^{\frac{n}{2}}$ distinct (up to complementation) affine function on $\frac{n}{2}$ variables.

Similar kind of concatenation technique has also been used for construction of resilient functions [24] (see also [155, 150]). One idea in this direction is to concatenate k-variable affine functions (repetition may be allowed) non degenerate on at least $m + 1$ variables to generate an m-resilient function f on n-variables. For such a function f, it is easy to find an annihilator g of degree $n - k + 1$ as described in [123]. In fact, it is shown in [38] that, unless a heavy condition is satisfied (which is very improbable unless k is almost equal to n), it is easy to find an annihilator of degree $n - k$. It has been commented in [123, Example 1 and the following paragraph] that k is generally greater than $\frac{n}{2}$ (this seems true for the Maiorana-McFarland type of functions presented in [135, 32]; but this has not been checked for some large classes of Maiorana-McFarland type of functions described in [150, 34]) and hence it is possible to get an annihilator g of degree less than $\frac{n}{2}$. However, it should be noted that in construction of resilient functions, there are lot of techniques [150] that use concatenation of k-variable affine functions where $k < \frac{n}{2}$. In such a case, the annihilators

described in [123, Theorem 2] will be of degree greater than $\frac{n}{2}$ and will not be of practical use as there are other annihilators of degree $\leq \frac{n}{2}$ which are not of the form given in [123, Theorem 2].

As example, the function H^0 in Example 5 above can be seen as concatenation of 3-variable affine functions $x_1+x_2, x_2+x_3, x_1+x_3, x_1+x_2+x_3$ non degenerate on at least two variables. In a similar fashion, the functions H^1, H^2, H^3 can also be seen as concatenation of only these four linear functions on 3-variables. Thus, it is clear that the assumption in the paper [123] that $k > \frac{n}{2}$ is not a valid assumption for $n \geq 8$ in this example.

We will show that even in such a case, Proposition 5 can provide further insight. In the next sub subsection we will show that a well known construction of resilient function [150, Theorem 10(b)] on n-variables (n odd) can never achieve the algebraic immunity $\lceil \frac{n}{2} \rceil$. At the best, it can only achieve the value $\lfloor \frac{n}{2} \rfloor$.

$(9, 1, 7, 240)$ Functions Constructed in [150]

We also like to present some interesting observations on $(9, 1, 7, 240)$ functions constructed in [150, Theorem 10(b)]. These functions can be seen as concatenation of affine functions on 3-variables, non degenerate on at least one variable. To explain this construction we briefly present some notations from [150].

Take a bit b and a bit string $s = s_0 \dots s_{n-1}$. Then the string b AND $s = s'_0 \dots s'_{n-1}$, where $s'_i = b$ AND s_i. Take two bit strings $x = x_0 \dots x_{n-1}$ and $y = y_0 \dots y_{m-1}$. The Kronecker product $x \otimes y = (x_0 \text{ AND } y) \dots (x_{n-1} \text{ AND } y)$, which is a string of length nm. The direct sum of two bit strings x, y is $x\$y = (x \otimes y^c) + (x^c \otimes y)$, where x^c, y^c are bitwise complement of x, y respectively. As an example presented in [150], if $f = 01$, and $g = 0110$, then $f\$g = 01101001$. Now we present the construction for $(2p+1, 1, 2p-1, 2^{2p}-2^p)$ function as presented in [150] for $p \geq 4$.

Construction 2 *[150, Theorem 10(b)] Let $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ be the 3-variable linear functions non degenerate on two variables (i.e., the functions $x_1+x_2, x_2+x_3, x_1+x_3, x_1+x_2+x_3$) and μ_1, μ_2, μ_3 be the 3-variable linear functions non degenerate on 1 variable (i.e., the functions x_1, x_2, x_3). Let g_i be the concatenation of the 3-variable function μ_i and its complement μ_i^c, for $1 \leq i \leq 3$. That is g_i's are basically 4-variable functions. Let h_1, h_2 be bent functions on $2p-4$ variables, and h_3, h_4, h_5 be bent functions of $2p-6$ variables and h_6, h_7 be two strings of lengths $2^{2p-6}+1$ and $2^{2p-6}-1$ which are prepared by properly adding and removing 1 bit from the truth table of $(2p-6)$-variable bent functions respectively. Let f be a concatenation*

of the following sequence of functions. $h_1\$\lambda_1, h_2\$\lambda_2, h_3\$g_1, h_4\$g_2, h_5\$g_3, h_6\$\lambda_3, h_7\$\lambda_4$. *This is a* $(2p+1, 1, 2p-1, 2^{2p}-2^p)$ *function.*

Proposition 7 *The* $(2p+1)$*-variable function presented in Construction 2 has a sub function of degree at most* $p-1$ *when* $x_{2p+1} = 0$.

Proof : Consider the sub function when $x_{2p+1} = 0$. The sub function (call it g) in concatenation form is $h_1\$\lambda_1, h_2\$\lambda_2$. Since h_1, h_2 are bent functions on $2p-4$ variables, they can have algebraic degree at most $p-2$. Further λ_1, λ_2 are 3-variable linear functions. The algebraic normal form of g is $(1+x_{2p})(h_1+\lambda_1)+x_{2p}(h_2+\lambda_2)$. So the degree of g is $\leq 1+(p-2) = p-1$. ■

Theorem 8 *For a function* $f \in \mathcal{B}_n$ *(n odd) generated out of Construction 2,* $\mathsf{AI}_n(f) \leq \lfloor \frac{n}{2} \rfloor$.

Proof : Here $n = 2p+1$. We take $g \in \mathcal{B}_{n-1}$, i.e., $r = 1$ according to Proposition 5. Further from Proposition 7, $\deg(g) \leq p-1 = \frac{n-1}{2} - 1$. Thus, $\mathsf{AI}_n(f) \leq \frac{n-1}{2} - 1 + 1 = \lfloor \frac{n}{2} \rfloor$. ■

Thus using our technique we can show that for odd n the construction proposed in [150, Theorem 10(b)] can not achieve the maximum possible algebraic immunity $\lceil \frac{n}{2} \rceil$. The maximum value it can achieve is $\leq \lfloor \frac{n}{2} \rfloor$. This can be seen only by Proposition 5 which generalizes the result of [123, Theorem 2, Example 1]. Here we give an example towards this. The function is constructed by Construction 2 for $p = 4$ and the functions constructed are as follows.

Example 6 *For* $p = 4$*, we choose the functions:*

1. $h_1 = 0000010100110110, h_2 = 0000010100110110, h_3 = 0001, h_4 = 0001, h_5 = 0001, h_6 = 00010, h_7 = 001$. *In this case, one gets a* $(9, 1, 7, 240)$ *function* f_1 *with* $\mathsf{AI}_9(f_1) = 3$.
2. *If one changes* $h_2 = 0000010100110110$ *by* $h_2 = 0000010100111001$*, then we get a* $(9, 1, 7, 240)$ *function* f_2 *with* $\mathsf{AI}_9(f_2) = 4$.

The question raised is why the algebraic immunity of these two functions are different? The reason is in the first case the functions h_1, h_2 are same with the ANF $x_1x_3 + x_2x_4$. Thus the sub function g (i.e., $h_1\$\lambda_1, h_2\$\lambda_2$) is a degree 2 function. So the maximum algebraic immunity, according to Proposition 5 can be $2+1 = 3$. That is the value achieved in Item 1. In the second case, h_1 is different from h_2 and the algebraic degree of g (i.e., $h_1\$\lambda_1, h_2\$\lambda_2$)

becomes 3 and it achieves the value $3 + 1 = 4$. Thus Proposition 5 helps in answering this question. It is important to note that this technique can be employed to study the upper bound of algebraic immunity for various constructions by analysing their sub functions and in particular, directly for the constructions proposed in [150, 32].

Based on the above discussion we like to make the following comments.

(1) There are Maiorana-McFarland type of constructions (concatenation of affine functions) where the concatenation of affine functions on small number of variables is exploited. In such a case, the annihilators presented in [123] will be not of much use. Thus in line of comments presented in [91], we too argue here that there is no reason to consider that the Maiorana-McFarland type constructions are inherently weak in terms of algebraic immunity.

(2) In Example 6, we note that changing the order of affine functions can change the algebraic immunity without any change in order of resiliency, nonlinearity and algebraic degree. The change in last four bits in h_2 implies that the concatenation of $\lambda_2, 1+\lambda_2, 1+\lambda_2, \lambda_2$ will be replaced by $1 + \lambda_2, \lambda_2, \lambda_2, 1 + \lambda_2$. This increases the algebraic immunity from 3 to 4. It will be of great interest to study the functions presented in [150, 151, 32].

3.5 Conclusion

In this chapter the algebraic immunity of a Boolean function is studied. We first identified a fundamental relationship between the Walsh spectrum and algebraic immunity of a Boolean function, leading to a lower bound on the nonlinearity. We followed with certain enumeration results of independent annihilators, which have some interest from cryptanalytic viewpoint. We then studied algebraic immunity of a Boolean functions in terms of the algebraic immunity of its sub functions. Further we also point out that functions having low degree sub functions are not good in terms of algebraic immunity and study some well known existing constructions from this approach. We have also studied some existing constructions in terms of their algebraic immunity, both theoretically and experimentally; this knowledge is necessary for practical design of cryptographic functions.

Chapter 4

First Construction of Boolean Functions having Optimal AI

Recent literature shows that algebraic attacks have gained a lot of attention in cryptanalysing stream and block cipher systems. So far very little attempt has been made to provide construction of Boolean functions primarily to resist certain kinds of algebraic attacks. In Chapter 3, Section 3.4, some existing construction methods have been analysed, that can provide Boolean functions with some other cryptographic properties, to see how good they are in terms of algebraic immunity. Algebraic immunity of certain constructions have also been studied in [16, 17, 18, 33].

So far there was no existing construction method that can achieve maximum possible algebraic immunity. For the first time, we provided a construction method where the algebraic immunity is the main concern. We showed that given a Boolean function on $n - 2d$ variables having algebraic immunity 1 or more, one can always construct a Boolean function on n variables with algebraic immunity at least $d+1$. The construction is iterative in nature (a function with two more variables is constructed in each step) and we need to apply it d times to get an n-variable function from an $(n-2d)$-variable initial function. The construction preserves the order of resiliency of the initial function and increases the nonlinearity by more than 2^{2d} times in d-steps (as it can be seen as a direct sum of a function with good nonlinearity and resiliency with another function with good algebraic immunity). Using our construction one can generate n-variable Boolean functions with highest possible algebraic immunity $\lceil \frac{n}{2} \rceil$.

In [38], it has been explained that one can achieve n-variable Boolean functions having algebraic immunity $\lfloor \frac{n}{2} \rfloor$ by random search. However, the maximum bound of algebraic immunity is $\lceil \frac{n}{2} \rceil$. Thus for even number of variables, random search provides Boolean functions having optimum algebraic immunity. However, optimal algebraic immunity can not be achieved for Boolean functions on odd number of variables by random search. We also like to point out that these theoretical developments in constructing Boolean functions with optimum algebraic immunity shed new light in further investigations in this area, which can not be supplemented by availability of Boolean functions with good algebraic immunity by random search.

4.1 Construction to Get Optimal AI

In this section we present a construction to get Boolean function ϕ_{2k} of $2k$ variables with algebraic immunity k. The construction is iterative in nature and it starts from an initial function $\phi_0 = 0$. In each step, 2 variables are added and algebraic immunity gets increased by 1. Let us now formalize the construction.

Construction 3 *Denote by $\phi_{2k} \in \mathcal{B}_{2k}$ the function defined by the recursion:*

$$\phi_{2k+2} = \phi_{2k}||\phi_{2k}||\phi_{2k}||\phi^1_{2k}, \tag{4.1}$$

where $||$ denotes the concatenation of the truth tables. In terms of algebraic normal form, $\phi_{2k+2} = \phi_{2k} + x_{2k+1}x_{2k+2}(\phi_{2k} + \phi^1_{2k})$, and where ϕ^1_{2k} is defined itself by a doubly indexed recursion

$$\phi^i_{2j} = \phi^{i-1}_{2j-2}||\phi^i_{2j-2}||\phi^i_{2j-2}||\phi^{i+1}_{2j-2}, \tag{4.2}$$

i.e., in terms of ANF, $\phi^i_{2j} = \phi^{i-1}_{2j-2} + (x_{2j-1} + x_{2j})(\phi^{i-1}_{2j-2} + \phi^i_{2j-2}) + x_{2j-1}x_{2j}(\phi^{i-1}_{2j-2} + \phi^{i+1}_{2j-2})$ for $j > 0$, $i > 0$, with base step $\phi^0_j = \phi_j$ for $j > 0$, $\phi^i_0 = i \bmod 2$ for $i \geq 0$.

To understand the recursion in the Construction 3, we present an example up to some depth.

- $\phi^1_{2k} = \phi_{2k-2}||\phi^1_{2k-2}||\phi^1_{2k-2}||\phi^2_{2k-2}$.
- $\phi^2_{2k-2} = \phi^1_{2k-4}||\phi^2_{2k-4}||\phi^2_{2k-4}||\phi^3_{2k-4}$.
- $\phi^3_{2k-4} = \phi^2_{2k-6}||\phi^3_{2k-6}||\phi^3_{2k-6}||\phi^4_{2k-6}$.

This goes on until we reach the null level for at least one of the two indices. Below we present the construction idea as truth table concatenation.

Step 1: $\phi_0 =$ 0
Step 2: $\phi_2 =$ 0001
Step 3: $\phi_4 =$ $\phi_2\phi_2\phi_2 0110$
Step 4: $\phi_6 =$ $\phi_4\phi_4\phi_4\phi_2 011001101001$
Step 5: $\phi_8 =$ $\phi_6\phi_6\phi_6\phi_4\phi_2 011001101001\phi_2 01100110100101101001100110010110$

What we actually prove now is the minimum degree annihilators of $f + \phi_{2k}$ are at the degree greater than k for any nonzero function f and from this we deduce the minimum degree annihilators of ϕ_{2k} and $1 + \phi_{2k}$ are at the degree k and $k + 1$ respectively.

To prove that for a nonzero f, $f + \phi_{2k}$ has algebraic immunity greater than k, we need some intermediate results. In the proofs, we will use the fact that, for any $F \in \mathcal{B}_n$ and any subset V of $\mathbb{F}_2^n$, the restriction to V of an annihilator of F is an annihilator of the restriction of F to V. For technical reasons, during our proofs, we will encounter certain situations when the degree of a function is negative. As such functions cannot exist, we will replace those functions by function 0.

Lemma 2 *Assume that the function $\phi_{2i} \in \mathcal{B}_{2i}$ has been generated by Construction 3 for $0 \leq i \leq k$ and $f + \phi_{2i}$ has no annihilator of degree less than $i + 1$ for $0 \leq i \leq k$ and f is a nonzero function of other variables. If, for some $0 \leq i \leq k$ and $j \geq 0$, there exist $g \in AN(f + \phi_{2i}^j)$ and $h \in AN(f + \phi_{2i}^{j+1})$ such that $\deg(g + h) \leq i - 1 - j$ then $g = h$.*

Proof : We prove the lemma by induction on i.

For the base step $i = 0$, $\deg(g + h) \leq 0 - 1 - j \leq -1$ implies that such a function cannot exist, i.e., $g + h$ is identically 0, which gives $g = h$.

Now we prove the inductive step. Assume that, for $i < \ell$, the induction assumption holds (for every $j \geq 0$). We will show it for $i = \ell$ (and for every $j \geq 0$). Suppose that there exist $g \in AN(f + \phi_{2\ell}^j)$ and $h \in AN(f + \phi_{2\ell}^{j+1})$ with $\deg(g + h) \leq \ell - 1 - j$. By construction, if $j > 0$ then we have

$$\begin{aligned} \phi_{2\ell}^j &= \phi_{2(\ell-1)}^{j-1} || \phi_{2(\ell-1)}^j || \phi_{2(\ell-1)}^j || \phi_{2(\ell-1)}^{j+1}, \\ \phi_{2\ell}^{j+1} &= \phi_{2(\ell-1)}^j || \phi_{2(\ell-1)}^{j+1} || \phi_{2(\ell-1)}^{j+1} || \phi_{2(\ell-1)}^{j+2}, \end{aligned}$$

and if $j = 0$ then

$$\phi_{2\ell}^0 = \phi_{2(\ell-1)}^0 || \phi_{2(\ell-1)}^0 || \phi_{2(\ell-1)}^0 || \phi_{2(\ell-1)}^1.$$

Let us denote

$$\begin{aligned} g &= v_1||v_2||v_3||v_4, \\ h &= v_5||v_6||v_7||v_8. \end{aligned}$$

Since $\deg(g+h) \leq \ell - 1 - j$, from the ANF of $g + h = (v_1 + v_5) + x_{2\ell-1}(v_1 + v_5 + v_2 + v_6) + x_{2\ell}(v_1 + v_5 + v_3 + v_7) + x_{2\ell-1}x_{2\ell}(v_1 + \cdots + v_8)$ we deduce the following:

- $\deg(v_1 + v_5) \leq \ell - 1 - j = (\ell - 1) - 1 - (j - 1)$. If $j > 0$ then $v_1 \in AN(f + \phi^{j-1}_{2(\ell-1)}), v_5 \in AN(f + \phi^{j}_{2(\ell-1)})$ implies that $v_1 = v_5$, according to the induction assumption. If $j = 0$, then we have $v_1, v_5 \in AN(f + \phi_{2(\ell-1)})$, and therefore $(v_1 + v_5) \in AN(f + \phi_{2(\ell-1)})$, with $\deg(v_1 + v_5) \leq \ell - 1$. Suppose that $v_1 + v_5 \neq 0$, then we would have $\deg(v_1 + v_5) \geq \ell$, since $f + \phi_{2(\ell-1)})$ has no annihilator of degree $\leq \ell - 1$, by hypothesis; a contradiction. Hence $v_1 + v_5 = 0$, i.e., $v_1 = v_5$.

- $\deg(v_2 + v_6) \leq (\ell - 1) - 1 - j$ and $v_2 \in AN(f + \phi^{j}_{2(\ell-1)}), v_6 \in AN(f + \phi^{j+1}_{2(\ell-1)})$, imply that $v_2 = v_6$, according to the induction assumption.

- $\deg(v_3 + v_7) \leq (\ell - 1) - 1 - j$ and $v_3 \in AN(f + \phi^{j}_{2(\ell-1)}), v_7 \in AN(f + \phi^{j+1}_{2(\ell-1)})$, imply that $v_3 = v_7$, according to the induction assumption.

- $\deg(v_4 + v_8) \leq (\ell - 1) - 1 - (j + 1)$ and $v_4 \in AN(f + \phi^{j+1}_{2(\ell-1)}), v_8 \in AN(f + \phi^{j+2}_{2(\ell-1)})$, imply that $v_4 = v_8$, according to the induction assumption.

Hence we get $g = h$. ■

Lemma 3 *Assume that the function $\phi_{2i} \in \mathcal{B}_{2i}$ has been generated by Construction 3 for $0 \leq i \leq k$ and that $f + \phi_{2i}$ has no annihilator of degree less than $i + 1$ for $0 \leq i \leq k$ where f is a nonzero function other variables. If, for some $0 \leq i \leq k$ and $j \geq 0$, there exists $g \in AN(f + \phi^{j}_{2i}) \cap AN(f + \phi^{j+1}_{2i})$ such that $\deg(g) \leq i + j + 1$, then $g = 0$.*

Proof : We prove the lemma by induction on $i - j$. For the base step (i.e., $i - j \leq 0$), we have from Construction 3 $f + \phi^{j+1}_{2i} = 1 + f + \phi^{j}_{2i}$ (this can easily be checked by induction). Hence, $g \in AN(f + \phi^{j}_{2i}) \cap AN(f + \phi^{j}_{2i} + 1)$, and $g = 0$.

Now we prove the inductive step. Assume that the induction assumption holds for $i - j \leq \ell$, $\ell \geq 0$, and let us prove it for $i - j = \ell + 1$. So let $g \in AN(f + \phi^{j}_{2i}) \cap AN(f + \phi^{j+1}_{2i})$

where $i-j=\ell+1$. If $j>0$, we have

$$\begin{aligned}\phi_{2i}^{j} &= \phi_{2(i-1)}^{j-1}||\phi_{2(i-1)}^{j}||\phi_{2(i-1)}^{j}||\phi_{2(i-1)}^{j+1},\\ \phi_{2i}^{j+1} &= \phi_{2(i-1)}^{j}||\phi_{2(i-1)}^{j+1}||\phi_{2(i-1)}^{j+1}||\phi_{2(i-1)}^{j+2}.\end{aligned}$$

Let us denote

$$g = v_1||v_2||v_3||v_4,$$

where, $v_1 \in AN(f+\phi_{2(i-1)}^{j-1}) \cap AN(f+\phi_{2(i-1)}^{j})$, $v_2, v_3 \in AN(f+\phi_{2(i-1)}^{j}) \cap AN(f+\phi_{2(i-1)}^{j+1})$ and $v_4 \in AN(f+\phi_{2(i-1)}^{j+1}) \cap AN(f+\phi_{2(i-1)}^{j+2})$.

1. Since $\deg(g) \leq i+j+1$, we have $\deg(v_4) \leq i+j+1 = (i-1)+(j+1)+1$. Since $(i-1)-(j+1) = i-j-2 < \ell$, we have $v_4 = 0$, according to the induction assumption. So the ANF of g is $v_1+x_{2i-1}(v_1+v_2)+x_{2i}(v_1+v_3)+x_{2i-1}x_{2i}(v_1+v_2+v_3)$. Then $\deg(v_1+v_2), \deg(v_1+v_3), \deg(v_1+v_2+v_3) \leq i+j$, which implies $\deg(v_1), \deg(v_2), \deg(v_3) \leq i+j$.

2. We have then $\deg(v_2) \leq i+j = (i-1)+j+1$ and $\deg(v_3) \leq i+j = (i-1)+j+1$. Since $(i-1)-j = i-j-1 \leq \ell$, we have $v_2 = v_3 = 0$, according to the induction assumption.

3. Since $v_2 = v_3 = v_4 = 0$, the ANF of g is $(1+x_{2i-1}+x_{2i}+x_{2i-1}x_{2i})v_1$. So, $\deg(v_1) \leq i+j-1 = (i-1)+(j-1)+1$. Here $(i-1)-(j-1) = \ell+1$. So, we can not use the induction assumption directly. Now we break $\phi_{2(i-1)}^{j-1}, \phi_{2(i-1)}^{j}$ and v_1 again into four parts as

$$\begin{aligned}\phi_{2(i-1)}^{j-1} &= \phi_{2(i-2)}^{j-2}||\phi_{2(i-2)}^{j-1}||\phi_{2(i-2)}^{j-1}||\phi_{2(i-2)}^{j},\\ \phi_{2(i-1)}^{j} &= \phi_{2(i-2)}^{j-1}||\phi_{2(i-2)}^{j}||\phi_{2(i-2)}^{j}||\phi_{2(i-2)}^{j+1},\\ v_1 &= v_{1,1}||v_{1,2}||v_{1,3}||v_{1,4}.\end{aligned}$$

   Using similar arguments as in Items 1 and 2, we have $v_{1,2} = v_{1,3} = v_{1,4} = 0$. So, $\deg(v_{1,1}) \leq i+j-3$. Doing the similar process j times, we will get some function $v \in AN(f+\phi_{2(i-j)}) \cap AN(f+\phi_{2(i-j)}^{1})$. At every step of this sub-induction, the degree decreases by 2, and we have then $\deg(v) \leq i+j+1-2j = i-j+1$. Breaking $\phi_{2(i-j)}, \phi_{2(i-j)}^{1}$ and v a last time into four parts and using that $v \in AN(f+\phi_{2(i-j)}) \cap AN(f+\phi_{2(i-j)}^{1})$, we have

$$\begin{aligned}\phi_{2(i-j)} &= \phi_{2(i-j-1)}||\phi_{2(i-j-1)}||\phi_{2(i-j-1)}||\phi_{2(i-j-1)}^{1},\\ \phi_{2(i-j)}^{1} &= \phi_{2(i-j-1)}||\phi_{2(i-j-1)}^{1}||\phi_{2(i-j-1)}^{1}||\phi_{2(i-j-1)}^{2},\\ v &= v'||v''||v'''||v''''.\end{aligned}$$

Using similar arguments as in Items 1 and 2, we have $v'' = v''' = v'''' = 0$. So, $\deg(v') \leq i - j - 1$. And $v' \in AN(f + \phi_{2(i-j-1)})$ implies that, if $v' \neq 0$, then $\deg(v) \geq i - j$, a contradiction. Hence, $v' = 0$ which implies $g = 0$.

If $j = 0$, then the proof is similar to the last step in Item 3 above. ■

Theorem 9 *Let $f' \in \mathcal{B}_{l+2k} = f + \phi_{2k}$ where $f \in \mathcal{B}_l$ is a non zero function depends on variables $\{x_1, x_2, \ldots, x_l\}$ and $\phi_{2k} \in \mathcal{B}_{2k}$ depends on variables $\{x_{l+1}, x_{l+2}, \ldots, x_{2k+l}\}$ for $k, l \geq 0$. Then f' has no annihilator of degree less than $k + 1$.*

Proof : We prove it by induction on k. For $k = 0$, we have $f' = f$ and hence there is no annihilator of degree less than 1. In the inductive step, we assume the hypothesis true until k and we have to prove that any nonzero function $g_{2k+2} \in \mathcal{B}_{l+2k+2}$ such that $f' * g_{2k+2} = 0$ has degree at least $k + 2$ where $f' = f + \phi_{2k+2}$. Suppose that such a function g_{2k+2} with degree less than or equal to $k + 1$ exists. Then, fixing the variables x_{l+2k+1} and x_{l+2k+2} the truth table of g_{2k+2} can be decomposed as

$$g_{2k+2} = g_{2k}||g'_{2k}||g''_{2k}||h_{2k},$$

where $g_{2k}, g'_{2k}, g''_{2k} \in AN(f + \phi_{2k})$, and $h_{2k} \in AN(f + \phi^1_{2k})$. The algebraic normal form of g_{2k+2} is then $g_{2k+2}(x_1, x_2, \ldots, x_{l+2k+2}) = g_{2k} + x_{l+2k+1}(g_{2k} + g'_{2k}) + x_{l+2k+2}(g_{2k} + g''_{2k}) + x_{l+2k+1}x_{l+2k+2}(g_{2k} + g'_{2k} + g''_{2k} + h_{2k})$.

If $\deg(g_{2k+2}) \leq k + 1$, then $\deg(g_{2k} + g'_{2k}) \leq k$ and $\deg(g_{2k} + g''_{2k}) \leq k$. Because both functions lie in $AN(f + \phi_{2k})$ and according induction assumption $f + \phi_{2k}$ has no annihilator of degree less than $k + 1$, we deduce that $g_{2k} + g'_{2k} = 0$ and $g_{2k} + g''_{2k} = 0$, which give, $g_{2k} = g'_{2k} = g''_{2k}$. Therefore, $g_{2k+2} = g_{2k} + x_{2k+1}x_{2k+2}(g_{2k} + h_{2k})$, $\deg(g_{2k}) \leq k + 1$ and $\deg(g_{2k} + h_{2k}) \leq k - 1$. According to Lemma 2, we have $g_{2k} = h_{2k}$ which implies $g_{2k} \in AN(f + \phi_{2k}) \cap AN(f + \phi^1_{2k})$. According to Lemma 3, we have then $g_{2k} = h_{2k} = 0$ that gives, $g_{2k+2} = 0$. This completes the proof. ■

Remark 1 *If $f \in \mathcal{B}_l$ (in above Theorem 9) has no annihilator of degree less than t where $t \geq 2$, then the question is whether $f + \phi_{2k}$ has no annihilator of degree less than $t + k$. In general, the answer is no. Because in Lemma 2 we have to consider $\deg(g+h) \leq i-2-j+t$ and in the base step in the proof of the lemma, i.e., for $i = 0$, $\deg(g + h) \leq -2 - j + t$. So for $j = 0$, $\deg(g + h) \leq t - 2$ where $t - 2 \geq 0$. So, we can not tell that $g + h = 0$. So, it is always true for the case $t \leq 1$, but not for $t \geq 2$.*

4.2 Cryptographic Properties of the Constructed Function ϕ_{2k}

In this section we study some important cryptographic properties like weight, nonlinearity, resiliency etc. of ϕ_{2k}.

Corollary 7

1. *$1 + \phi_{2k}$ has no annihilator of degree less than $k + 1$.*
2. *ϕ_{2k} has no annihilator of degree less than k.*
3. *ϕ_{2k} and $1 + \phi_{2k}$ have annihilators of degree k and $k + 1$ respectively, i.e., the lowest degree annihilators of ϕ_{2k} and $1 + \phi_{2k}$ are at degree k and $k + 1$ respectively.*

Proof : The proof of item 1 directly follows from Theorem 9 by taking $f \in \mathcal{B}_0$ is constant 1 function, i.e., the truth table of f contains a single 1. As f is nonzero, following Theorem 9, $1 + \phi_{2k}$ has no annihilator of degree less than or equal to k.

Now we prove item 2. Consider the function $f' = x_1 + \phi_{2k}$, i.e., where the indeterminates of ϕ_{2k} are $x_2, x_3, \ldots, x_{2k+1}$. Following the Theorem 9 we have both f' and $1 + f'$ have no annihilator of degree less than $k+1$. That is $\mathsf{AI}(f') = k+1$, maximum value. Now following the item 1 in Corollary 6 of Chapter 3 in contrapositive way we have $\mathsf{AI}(\phi_{2k}) = k$, i.e., maximum value. So, ϕ_{2k} has no annihilator of degree less than k.

Now we prove item 3. Since ϕ_{2k} and $1 + \phi_{2k}$ have no annihilator of degree less than k and $k + 1$ respectively, following the lines of the proof of the Theorem 2 of Chapter 3 we have $\mathsf{wt}(\phi_{2k}) \geq \sum_{i=0}^{k-1} \binom{n}{i}$ and $\mathsf{wt}(1 + \phi_{2k}) \geq \sum_{i=0}^{k} \binom{n}{i}$. As $sup(\phi_{2k}) \cup sup(1 + \phi_{2k}) = \mathbb{F}_2^{2k}$ and $|\mathbb{F}_2^{2k}| = 2^{2k}$, we have $\mathsf{wt}(\phi_{2k}) = \sum_{i=0}^{k-1} \binom{n}{i}$ and $\mathsf{wt}(1 + \phi_{2k}) = \sum_{i=0}^{k} \binom{n}{i}$. Which implies ϕ_{2k} and $1 + \phi_{2k}$ have annihilators of degree k and $k + 1$ respectively. ■

Apart from ϕ_n, using this Construction 3, one can generate functions of n variable whose algebraic immunity is the highest possible, i.e., $\lceil \frac{n}{2} \rceil$. $\mathsf{AI}(f + \phi_n) = \frac{n}{2} + 1$, where f is an 1 or 2-variable non constant function. Note that the algebraic immunity stays the same if a function is subjected to affine transformation on input variables. Thus, taking any function presented in the above discussion, one can apply affine transformation to get number of functions. Further the nonlinearity and algebraic degree also stays same after affine transformation.

Now we will discuss some cryptographic properties of the function $\phi_{2k},\ k > 0$ generated using Construction 3.

Proposition 8 *Let $\phi_{2k},\ k > 0$ be constructed using Construction 3. Then*

1. $\mathsf{wt}(\phi_{2k}) = \sum_{i=0}^{k-1} \binom{2k}{i} = 2^{2k-1} - \binom{2k-1}{k}$.
2. $\mathsf{nl}(\phi_{2k}) = \sum_{i=0}^{k-1} \binom{2k}{i} = 2^{2k-1} - \binom{2k-1}{k}$.
3. $\mathsf{AI}(\phi_{2k}) = k$.

Proof : In the proof for Items 1 and 3 comes from the lines of the proof of Corollary 7. To prove Item 2, we have $\mathsf{nl}(\phi_{2k}) \leq \mathsf{wt}(\phi_{2k}) = 2^{2k-1} - \binom{2k-1}{k}$. Following the Theorem 4 of Chapter 3, we have $\mathsf{nl}(x_{2k+1}+\phi_{2k}) \geq 2^{2k} - \binom{2k}{k}$ as $\mathsf{AI}(x_{2k+1}+\phi_{2k}) = k+1$. As $\mathsf{nl}(x_{2k+1}+\phi_{2k}) = 2\mathsf{nl}(\phi_{2k})$, we get $\mathsf{nl}(\phi_{2k}) \geq 2^{2k-1} - \frac{1}{2}\binom{2k}{k} = 2^{2k-1} - \binom{2k-1}{k}$. ■

The following result related to algebraic degree of ϕ_{2k} is due to Carlet [36, 37].

Proposition 9 *For $k \geq 1$ the degree of ϕ_{2k} is as follows:*

1. $\deg(\phi_{2k}) = 2k$ *if and only if k is a power of* 2.
2. *If neither k nor $k+1$ is a power of* 2, *then* $\deg(\phi_{2k}) = 2k - 1$.
3. *If $k+1$ is a power of* 2, *then* $2k - 3 \leq \deg(\phi_{2k}) \leq 2k - 1$.

Further, we will discuss some cryptographic properties of the functions $f_{l+2k} = f_l + \phi_{2k}$ where f_l is a nonzero function depends on $x_1, \ldots, x_l$ and ϕ_{2k} depends on $x_{l+1}, \ldots, x_{l+2k}$ using Construction 3.

Corollary 8 *Let $f_l \in \mathcal{B}_l$ be some l-variable nonzero function and let $f_{l+2k} = f_l + \phi_{2k}$, be the direct sum of f_l and ϕ_{2k} (i.e., the variable sets for f_l and ϕ_{2k} are disjoint). Then we have the following results.*

1. $\mathsf{nl}(f_{l+2k}) = 2^l\mathsf{nl}(\phi_{2k}) + 2^{2k}\mathsf{nl}(f_l) - 2\mathsf{nl}(\phi_{2k})\mathsf{nl}(f_l) > 4^k\mathsf{nl}(f_l)$.
2. *If f_l is r-resilient, then f_{l+2k} is also r-resilient.*
3. $\deg(f_{l+2k}) = \max\{\deg(f_l), \deg(\phi_{2k})\}$.
4. $wt(f_{l+2k}) = wt(\phi_{2k})(2^l - wt(f_l)) + (2^{2k} - wt(\phi_{2k}))wt(f_l)$.

Proof : The proof of item 1 follows from [150, Proposition 1(d)] and the proof of item 2 follows from [150, Proposition 1(c)]. The result related to algebraic degree and weight is also easy to see. ■

In particular, if f_l is a non-constant 1-variable function, we get a balanced function with optimum algebraic immunity. The function ϕ_{2k} can be computed very efficiently with linear time (see Item 3 of Section V-A in [36] and [37]). The number of elementary operations which have to be performed for calculating the output is less than $12k$.

4.3 Different Initializations on ϕ_{2k}

A drawback of the function ϕ_{2k} is that it is unbalanced. This happens since $\phi_2 = x_1x_2$ is unbalanced. If one starts the construction with ϕ_2 as affine function, then the function ϕ_{2k} will always be balanced as $\phi^i_{2j} = x_1 + x_2 + (i + j) \bmod 2$ for $i > 0, j > 0$. Now we present some observations in this regard.

1. Take $\phi_2 = x_1 + x_2$.

   Case 1: $\phi^i_2 = x_1 + x_2$ if i is even and $\phi^i_2 = 1 + x_1 + x_2$ if i is odd for $i > 0$. These are presented in the following table.

   Case 2: Also in brackets, we present the results when $\phi^i_2 = x_1 + x_2$ if i is odd and $\phi^i_2 = 1 + x_1 + x_2$ if i is even for $i > 0$.

function	degree	nonlinearity	resiliency	$\mathcal{AI}$
ϕ_2	1(1)	0(0)	1(1)	1(1)
ϕ_4	2(1)	4(0)	1(1)	2(1)
ϕ_6	4(4)	20(4)	1(1)	3(2)
ϕ_8	5(6)	88(28)	1(1)	4(3)
ϕ_{10}	8(8)	372(148)	1(1)	5(4)

   Here, for the first case, ϕ_{2k} is always 1-resilient, optimal algebraic immunity is achieved and nonlinearity is slightly lesser than what we have observed for Construction 3. However, in the second case, ϕ_{2k} has poor nonlinearity and lower AI.

2. Then we have attempted $\phi_2 = x_1$ and $\phi^i_2 = x_1 + x_2$ when i is even (respectively odd) and $\phi^i_2 = 1 + x_1 + x_2$ when i is odd (respectively even). We found algebraic immunity is optimal but poor nonlinearity. The results are same for both the cases so we do not write them separately in brackets.

function	degree	nonlinearity	resiliency	$\mathcal{AI}$
ϕ_2	1	0	0	1
ϕ_4	3	2	0	2
ϕ_6	4	12	0	3
ϕ_8	7	58	0	4
ϕ_{10}	8	260	0	5

3. Take $\phi_2 = x_1$ and $\phi_2^i = x_1 + x_2, i > 0$. We find that the ANF of ϕ_{2k} is of the form $\phi_{2k} = x_1 + x_2F$, where F is a function on $2k-2$ many variables. So, AI will be ≤ 2, since $(1+x_1)(1+x_2)$ is annihilator of ϕ_{2k} for any $k > 1$.

So it seems that just by changing the initializations in Construction 3, it may not be possible to get dramatically better results. One may need to attempt for completely different kinds of construction to achieve better parameters.

4.4 Conclusion

In this chapter we present a construction where one can get Boolean functions ϕ_{2k} with maximum possible algebraic immunity. This is the first construction in literature which provides optimal AI value. We studied some other cryptographic properties of the function ϕ_{2k}. The constructed functions have high degrees but are not balanced and have insufficient nonlinearity. However, they can be used in secondary constructions settling these drawbacks; the construction can be used in conjunction with Boolean functions with other cryptographic properties to have functions which are suitable for different cryptographic applications. We have also studied the behavior of the functions in terms of algebraic immunity by changing the initializations to get balanced functions.

Chapter 5

Basic Theory to construct Boolean functions of Optimal AI

Though there are increasing interest in construction of Boolean functions with good algebraic immunity [16, 17, 18, 33], so far there is only one construction method that can achieve the maximum possible algebraic immunity $\lceil \frac{n}{2} \rceil$ for an n-variable function proposed by us as described in Chapter 4. The heart of the construction in Chapter 4 is a function ϕ_{2k} on even $(2k)$ number of variables with maximum possible algebraic immunity k.

In this chapter we explain a generic construction idea of functions with maximum algebraic immunity that comes from the basic theory. By basic theory we mean Construction 4, Lemma 4 and Lemma 5 in Section 5.1 as this presents the concrete construction idea of Boolean functions with full algebraic immunity. We apply this basic theory to get symmetric functions with maximum possible algebraic immunity. For even n the weight and nonlinearity is $2^{n-1} - \binom{n-1}{\frac{n}{2}}$ and the algebraic degree is $(2^{\lfloor \log_2 n \rfloor})$.

For n even, we also provide a large class of balanced Boolean functions (not symmetric) with maximum possible algebraic immunity having nonlinearity $\geq 2^{n-1} - \binom{n}{\frac{n}{2}}$. Under experimental set up, with a simple heuristic, we show that actually one can achieve much better nonlinearity than this lower bound (in fact very close to $2^{n-1} - \binom{n-1}{\frac{n}{2}}$). For odd n our construction provides symmetric balanced functions with nonlinearity $2^{n-1} - \binom{n-1}{\frac{n-1}{2}}$ and algebraic degree $(2^{\lfloor \log_2 n \rfloor})$.

As our basic construction starts from symmetric Boolean functions. The Walsh spectra of such functions are related to Krawtchouk Polynomials and we use these properties to get related results.

It is well known that the algebraic immunity (also algebraic degree and nonlinearity) of a Boolean function is invariant under affine transformation on the input variables. Thus one can easily apply affine transformation to get a wider class of functions (which are not symmetric) from our construction achieving the maximum possible algebraic immunity (with same algebraic degree and nonlinearity). It should be mentioned at this point that the non symmetry achieved by affine transformation will not provide any additional cryptographic strength to the functions, this is only to mention the large class of Boolean functions with full algebraic immunity.

Our basic construction provides symmetric Boolean functions. Referring [30], we like to add that though symmetric functions are well studied in many applications for their concise representation, symmetric functions with good cryptographic properties are yet to be exhibited. The other cryptographic properties (e.g., nonlinearity, correlation immunity etc.) of the Boolean functions (whether symmetric or not) that we consider here are not very good. Thus, in no way, we are proposing these functions for direct use in cryptosystems. The motivation of this chapter is systematic theoretical study of Boolean functions with maximum possible algebraic immunity.

5.1 Construction Using the Basic Theory

The idea of our construction comes from the following.

Construction 4 *Let $f, f_1, f_2 \in \mathcal{B}_n$ with the following conditions.*

1. *There is no annihilator of f_1, f_2 having degree less than $\lceil \frac{n}{2} \rceil$.*
2. *$supp(f) \supseteq supp(f_2)$ and $supp(1+f) \supseteq supp(f_1)$.*

Then we have the following important result.

Lemma 4 *Let $f \in \mathcal{B}_n$ be a function as described in Construction 4. Then $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$.*

Proof : As $supp(1+f) \supseteq supp(f_1)$, $AN(1+f) \subseteq AN(f_1)$ and as $supp(f) \supseteq supp(f_2)$, $AN(f) \subseteq AN(f_2)$. Since there is no annihilator of f_1, f_2 having degree less than $\lceil \frac{n}{2} \rceil$, neither f nor $1+f$ can have any annihilator of degree less than $\lceil \frac{n}{2} \rceil$. Thus $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$. ∎

Now we present the other direction.

Lemma 5 *Let $f \in \mathcal{B}_n$ and $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$. Then there exist $f_1, f_2 \in \mathcal{B}_n$ with $supp(f_1) \subseteq supp(1+f)$ and $supp(f_2) \subseteq supp(f)$ such that $\mathsf{wt}(f_1) = \mathsf{wt}(f_2) = \sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \binom{n}{i}$ and f_1, f_2 have no annihilator of degree less than $\lceil \frac{n}{2} \rceil$.*

Proof : Since $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$, f has no annihilator of degree less than $\lceil \frac{n}{2} \rceil$. That is, there cannot be any

$$g(x_1, \ldots, x_n) = a_0 + \sum_{i=0}^{n} a_i x_i + \cdots + \sum_{1 \leq i_1 \ldots \leq i_{\lceil \frac{n}{2} \rceil - 1} \leq n} a_{i_1 \ldots i_{\lceil \frac{n}{2} \rceil - 1}} x_{i_1} \cdots x_{i_{\lceil \frac{n}{2} \rceil - 1}}$$

such that $g(x_1, \ldots, x_n) = 0$ where $f(x_1, \ldots, x_n) = 1$. That is there is no nonzero solution of the system of homogeneous linear equations $g(x_1, \ldots, x_n) = 0$ for $(x_1, \ldots, x_n) \in supp(f)$ on a_i's, i.e., this system has full rank $(\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \binom{n}{i})$. So, there must be $\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \binom{n}{i}$ many linearly independent equations. Now we construct f_2 such that $supp(f_2)$ is the set of input vectors corresponding to $\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \binom{n}{i}$ many linearly independent equations. So, f_2 has no annihilator of degree less than $\lceil \frac{n}{2} \rceil$. Similarly, we can construct f_1 considering $(1+f)$ has no annihilator of degree less than $\lceil \frac{n}{2} \rceil$. ■

Based on Lemma 4 and Lemma 5, we get a clear idea of a construction strategy for a function with maximum possible algebraic immunity.

For odd n, there is no option other than $f_1 = f$ and $f_2 = 1 + f$ to have maximum algebraic immunity for f, since $\mathsf{wt}(f_1) + \mathsf{wt}(f_2) = 2 \sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} \binom{n}{i} = 2^n$. This fact also follows from Corollary 3 in Chapter 3 that a function on odd number of variables must be balanced (weight 2^{n-1} for n-variable function) to achieve the maximum possible algebraic immunity. Also recently it has been shown [26] that for balanced functions on odd number of variables, it is enough to consider the annihilators of f (the case for $1+f$ will automatically be deduced) in terms of maximum algebraic immunity. The exact result is as follows.

Proposition 10 [26] *Let $f \in \mathcal{B}_n$ (n odd) be balanced Boolean function and it does not have any annihilator with algebraic degree less than $\lceil \frac{n}{2} \rceil$. Then $1 + f$ has no annihilator with algebraic degree less than $\lceil \frac{n}{2} \rceil$. Consequently, $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$.*

However, for even n, $\mathsf{wt}(f_1)+\mathsf{wt}(f_2) = 2\sum_{i=0}^{\lceil \frac{n}{2} \rceil -1} \binom{n}{i} = 2^n - \binom{n}{\frac{n}{2}}$. So, a part of remaining $\binom{n}{\frac{n}{2}}$ output points can be chosen randomly to get different functions f without affecting the algebraic immunity. Hence for even n case this restriction is not as strict as odd n case.

5.1.1 A Construction for Maximum Algebraic Immunity

Let us now present the application of the basic theory for a concrete construction of functions having optimal algebraic immunity.

Construction 5 *Let $f \in \mathcal{B}_n$.*

1. *If n is odd then*

$$\begin{aligned} f(x_1,\ldots,x_n) &= 0 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) \leq \lfloor \tfrac{n}{2} \rfloor, \\ &= 1 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) \geq \lceil \tfrac{n}{2} \rceil. \end{aligned}$$

2. *If n is even then*

$$\begin{aligned} f(x_1,\ldots,x_n) &= 0 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) < \frac{n}{2}, \\ &= 1 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) > \frac{n}{2}, \\ &= b_{(x_1,\ldots,x_n)} \in \{0,1\} \text{ for } \mathsf{wt}(x_1,\ldots,x_n) = \frac{n}{2}. \end{aligned}$$

Lemma 6 *Define two functions $f_1, f_2 \in \mathcal{B}_n$ as follows.*

$$\begin{aligned} f_1(x_1,\ldots,x_n) &= 1 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) < \lceil \tfrac{n}{2} \rceil, \\ &= 0 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) \geq \lceil \tfrac{n}{2} \rceil. \end{aligned}$$

$$\begin{aligned} f_2(x_1,\ldots,x_n) &= 0 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) \leq \lfloor \tfrac{n}{2} \rfloor, \\ &= 1 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) > \lfloor \tfrac{n}{2} \rfloor. \end{aligned}$$

Then f_1, f_2 have no annihilator of degree less than $\lceil \frac{n}{2} \rceil$.

Proof : We first show that f_1 has no annihilator of degree less than $\lceil \frac{n}{2} \rceil$. Suppose f_1 has a nonzero annihilator $g \in \mathcal{B}_n$ having degree less than $\lceil \frac{n}{2} \rceil$ of the form

$$a_0 + \sum_{i=0}^{n} a_i x_i + \cdots + \sum_{1 \leq i_1 < \ldots < i_{\lceil \frac{n}{2} \rceil -1} \leq n} a_{i_1,\ldots,i_{\lceil \frac{n}{2} \rceil -1}} x_{i_1} \cdots x_{i_{\lceil \frac{n}{2} \rceil -1}},$$

where a's are in $\mathbb{F}_2$, but not all of them are zero. As g is an annihilator of f_1, $g(x_1, \ldots, x_n) = 0$ when $f_1(x_1, \ldots, x_n) = 1$. Hence solving the system of homogeneous linear equations (considering a's as the variables) formed by $g(x_1, \ldots, x_n) = 0$ when $f_1(x_1, \ldots, x_n) = 1$, we must get a nontrivial (not all zero) solution on a's.

Let us consider an input $(x_1, \ldots, x_n)$, where $x_{i_1}, \ldots, x_{i_t}$ are 1 ($t < \lceil \frac{n}{2} \rceil$) and the rest are 0 with $f_1(x_1, x_2, \ldots, x_n) = 1$. Then for this input, we have the homogeneous linear equation of the form

$$\sum_{I \subseteq \{i_1, \ldots, i_t\}} a_I = 0, \text{ i.e., } a_{i_1, \ldots, i_t} = \sum_{I \subset \{i_1, \ldots, i_t\}} a_I.$$

Since $f_1(0, \ldots, 0) = 1$, we must have $g(0, \ldots, 0) = 0$, i.e., $a_0 = 0$. As $f_1(x) = 1$ for $\mathsf{wt}(x) = 1$, we have $a_i = a_0 = 0$. Following the same process repeatedly till $\mathsf{wt}(x) = \lceil \frac{n}{2} \rceil - 1$, we have all a's in g are 0. Thus g becomes a zero function, which is a contradiction as we have started with nonzero g. Thus f_1 has no annihilator of degree less than $\frac{n}{2}$.

Now we show that f_2 has no annihilator of degree less than $\lceil \frac{n}{2} \rceil$. Suppose f_2 has an annihilator h of degree less than $\lceil \frac{n}{2} \rceil$. That is, $f_2(x_1, \cdots, x_n) * h(x_1, \cdots, x_n) = 0$. Note that $f_1(x_1, \cdots, x_n) = f_2(1 + x_1, \cdots, 1 + x_n)$, i.e., $f_2(x_1, \cdots, x_n) = f_1(1 + x_1, \cdots, 1 + x_n)$. Thus, $f_1(1+x_1, \cdots, 1+x_n) * h(x_1, \cdots, x_n) = 0$. Define h' as $h'(x_1, \cdots, x_n) = h(1+x_1, \cdots, 1+x_n)$, i.e., $h(x_1, \cdots, x_n) = h'(1 + x_1, \cdots, 1 + x_n)$. This gives $\deg(h') = \deg(h) < \lceil \frac{n}{2} \rceil$. Hence, we have $f_1(1+x_1, \cdots, 1+x_n) * h'(1+x_1, \cdots, 1+x_n) = 0$, i.e., $f_1(x_1, \ldots, x_n) * h'(x_1, \ldots, x_n) = 0$. So, f_1 has an annihilator of degree less than $\lceil \frac{n}{2} \rceil$, which is a contradiction. ■

Thus we get the following theorem.

Theorem 10 *Let $f(x_1, \ldots, x_n) \in \mathcal{B}_n$ constructed by Construction 5. Then $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$.*

Proof : First we prove for odd n. Here $supp(1 + f) = supp(f_1)$ and $supp(f) = supp(f_2)$, where f_1, f_2 are as described in Lemma 6. Thus from Lemma 4 we have the proof for odd n. Now we will prove for n even. It can be checked that $supp(1 + f) \supseteq supp(f_1)$ and $supp(f) \supseteq supp(f_2)$, where f_1, f_2 are as described in Lemma 6. This, using Lemma 4, gives the proof for n even. ■

5.2 Algebraic Degree and Nonlinearity for a Sub case

Here we can consider a special case of the functions in Construction 5 as follows:

Construction 6

$$\begin{aligned}\psi_n(x_1,\ldots,x_n) &= 0 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) \leq \lfloor \tfrac{n}{2} \rfloor, \\ &= 1 \text{ for } \mathsf{wt}(x_1,\ldots,x_n) > \lfloor \tfrac{n}{2} \rfloor.\end{aligned}$$

From the proof of the Lemma 6 we have the following corollary for even n.

Corollary 9

1. *ψ_{2k} and $1+\psi_{2k}$ has no annihilator of degree less than k and $k+1$ respectively.*
2. *ψ_{2k} and $1+\psi_{2k}$ have annihilators of degree k and $k+1$ respectively.*

Note that in this case ψ_n is a symmetric Boolean function (See Section 2.1.5 of Chapter 2 for definitions and terms). The function ψ_n is also called majority function as it is one for higher weight input vectors. Now we exactly calculate the algebraic degree, weight and nonlinearity of the functions in Construction 6.

Corollary 10 *For $k > 0$,*

1. $\mathsf{wt}(\psi_{2k}) = \sum_{i=0}^{k-1} \binom{2k}{i} = 2^{2k-1} - \binom{2k-1}{k}$.
2. $\mathsf{wt}(\psi_{2k+1}) = \sum_{i=0}^{k} \binom{2k+1}{i} = 2^{2k}$.

5.2.1 Algebraic Degree

The relationship between re_f, ra_f (see Section 2.1.5 of Chapter 2) for any symmetric function f have been presented in [117, Theorem 3] as

$$re_f(i) = (\sum_{k=0}^{i} ra_f(k)\binom{i}{k}) \bmod 2, \tag{5.1}$$

where $0 \leq i \leq n$. From [46, Page 85], for two integer sequences p, q,

$$p_i = \sum_{k=0}^{i} q_k \binom{i}{k} \text{ iff } q_i = \sum_{k=0}^{i} p_k (-1)^{i-k} \binom{i}{k}. \tag{5.2}$$

From Equation 5.1 and Equation 5.2 we get

Proposition 11 $ra_f(i) = (\sum_{k=0}^{i} re_f(k) \binom{i}{k}) \bmod 2$.

We have Lucas' theorem [45, page 79] as following.

Theorem 11 (Lucas' theorem) *Let two nonnegative integers a, b, written base p (prime) as $a = \sum_{i=0}^{e} a_i p^i$ and $b = \sum_{i=0}^{e} b_i p^i$ respectively, where $0 \leq a_i, b_i < p$. Then $\binom{a}{b} = \prod_{i=0}^{e} \binom{a_i}{b_i} \bmod p$. Consequently, if $p = 2$, $\binom{a}{b} = 1 \mod 2$ iff $supp(a) \supseteq supp(b)$.*

Now we have the following proposition comes from the Theorem 11.

Proposition 12 *Suppose n and k are nonnegative integers with $n \geq k$. Let $n = 2^t + l$ where $0 \leq l < 2^t$ and $t \geq 0$. Then we have*

1. *Let $k = 2^t + l_1$ where $l_1 \leq l$. Then $\binom{n}{k}$ is even iff $\binom{l}{l_1}$ is even.*
2. *Let $k = 2^{t-1} + l_2$ where $l_2 < 2^{t-1}$. Then $\binom{n}{k}$ is even if $k > l$.*

Proof : For item 1, we have $\binom{n}{k} = \binom{1}{1}\binom{l}{l_1} \mod 2$, i.e., $\binom{n}{k} = \binom{l}{l_1} \mod 2$. For item 2, we have $k = 2^{t-1} + l_2$ where $l_2 < 2^{t-1}$ and $l < k$ which implies $supp(l) \not\supseteq supp(k)$. So, $\binom{n}{k} = \binom{1}{0}\binom{l}{k} \mod 2$, i.e., $\binom{n}{k} = \binom{l}{k} \mod 2$, i.e., $\binom{n}{k} = 0 \mod 2$ as $supp(l) \not\subseteq supp(k)$. ■

The following result provides the algebraic normal form and degree of ψ_n.

Theorem 12 *Let $\psi_n \in \mathcal{B}_n$ a symmetric function as given in Construction 6. Then,*

1. $ra_{\psi_n}(i) = 0$ *for* $i \leq \lfloor \frac{n}{2} \rfloor$,
2. $ra_{\psi_n}(\lfloor \frac{n}{2} \rfloor + 1) = 1$,
3. $ra_{\psi_n}(i) = \sum_{k=\lfloor \frac{n}{2} \rfloor + 1}^{i} \binom{i}{k} \bmod 2$, *for* $i \geq \lfloor \frac{n}{2} \rfloor + 2$,
4. $\deg(\psi_n) = 2^{\lfloor \log_2 n \rfloor}$.

Proof : Given the function ψ_n, it is clear that $re_{\psi_n}(i) = 0$ for $0 \leq i \leq \lfloor \frac{n}{2} \rfloor$ and $re_{\psi_n}(i) = 1$ for $\lfloor \frac{n}{2} \rfloor + 1 \leq i \leq n$. Thus from $ra_{\psi_n}(i) = (\sum_{k=0}^{i} re_{\psi_n}(k)\binom{i}{k}) \bmod 2$ (Proposition 11), we get $ra_{\psi_n}(i) = 0$ for $i \leq \lfloor \frac{n}{2} \rfloor$ and $ra_{\psi_n}(\lfloor \frac{n}{2} \rfloor + 1) = 1$. So we get the proofs of items 1 and 2.

The item 3 follows from Proposition 11 considering the result from item 1 and using $re_{\psi_n}(k) = 1$ for $k \geq \lfloor \frac{n}{2} \rfloor + 1$.

Suppose $t = \lfloor \log_2 n \rfloor$ and $l = n - 2^t$, i.e., $n = 2^t + l$ where $0 \leq l < 2^t$ and $t \geq 0$. For item 4 we need to show that $ra_{\psi_n}(i) = 1$ for $i = 2^t = 2^{\lfloor \log_2 n \rfloor}$ and $ra_{\psi_n}(i) = 0$ for all $i > 2^t$. Now for $i \geq 2^t$, $ra_{\psi_n}(i) = \sum_{k=\lfloor \frac{n}{2} \rfloor + 1}^{i} \binom{i}{k} \bmod 2$. Here $n = 2^t + l$, i.e., $\lfloor \frac{n}{2} \rfloor + 1 = 2^{t-1} + \lfloor \frac{l}{2} \rfloor + 1$. Suppose $i = 2^t + l_1$ where $0 \leq l_1 \leq l$. Now for $\lfloor \frac{n}{2} \rfloor + 1 \leq k < 2^t$, we have $k \geq \lfloor \frac{n}{2} \rfloor + 1 = 2^{t-1} + \lfloor \frac{l}{2} \rfloor + 1 > l \geq l_1$ as $\lfloor \frac{l}{2} \rfloor < 2^{t-1}$. So following the fact $\binom{i}{k} = 0 \bmod 2$ for $\lfloor \frac{n}{2} \rfloor + 1 \leq k < 2^t$ in Proposition 12 (Item 2) we have $ra_{\psi_n}(i) = \sum_{k=2^t}^{i} \binom{i}{k} \bmod 2$. Then $ra_{\psi_n}(i) = \sum_{j=0}^{l_1} \binom{2^t+l_1}{2^t+j} \bmod 2$ as $i = 2^t + l_1$. Then following Proposition 12 (Item 1) we have $ra_{\psi_n}(i) = \sum_{j=0}^{l_1} \binom{l_1}{j} \bmod 2 = 2^{l_1} \bmod 2$. Thus, $ra_{\psi_n}(2^t) = 1$ as $l_1 = 0$ and $ra_{\psi_n}(i) = 0$ for $i > 2^t$ as $l_1 > 0$. ■

Let us explain this with an example of $\psi_9 \in \mathcal{B}_9$, a symmetric function, as given in Construction 6. In this case re_{ψ_9} will be of the form "0 0 0 0 0 1 1 1 1 1". The ra_{ψ_9} of this function will be "0 0 0 0 0 1 1 1 1 0". Thus the function will be of algebraic degree 8 and algebraic immunity 5.

5.2.2 Nonlinearity

In this sub section we will analyse the nonlinearity of the function ψ_n as explained in Construction 6. Nonlinearity is one of the most important cryptographic properties of Boolean functions which is used in cryptosystems to prevent linear attacks [73]. Moreover, this property is also very interesting from combinatorial point of view.

As the function ψ_n explained in Construction 6 is a symmetric Boolean function, we here concentrate on the Walsh spectra of this class. The Walsh spectra of symmetric Boolean functions have very nice combinatorial properties related to Krawtchouk polynomial [152].

Krawtchouk polynomial [112, Page 151, Part I] of degree i is given by

$$K_i(x, n) = \sum_{j=0}^{i} (-1)^j \binom{x}{j} \binom{n-x}{i-j}, \ i = 0, 1, \ldots, n. \tag{5.3}$$

It is known that for a fixed $\omega \in \mathbb{F}_2^n$, such that $\mathsf{wt}(\omega) = k$,

$$\sum_{x \in \mathbb{F}_2^n, \mathsf{wt}(x)=i} (-1)^{\omega \cdot x} = K_i(k, n).$$

Thus it can be checked that if $f \in \mathcal{B}_n$ is symmetric, then for $\mathsf{wt}(\omega) = k$,

$$W_f(\omega) = \sum_{i=0}^{n} (-1)^{re_f(i)} K_i(k, n). \tag{5.4}$$

This also implies that for a symmetric function $f \in \mathcal{B}_n$ and $\alpha, \beta \in \mathbb{F}_2^n$, $W_f(\alpha) = W_f(\beta)$, if $\mathsf{wt}(\alpha) = \mathsf{wt}(\beta)$. Thus it is enough to calculate the Walsh spectra for the inputs of $n+1$ different weights. Keeping this in mind, given a symmetric Boolean function $f \in \mathcal{B}_n$, we denote $rw_f(i) = W_f(\omega)$, such that $\mathsf{wt}(\omega) = i$. Thus rw_f can be seen as a mapping from $\{0, \ldots, n\}$ to $\mathbb{Z}$.

Let us now list some known results in this area [112, 106].

Proposition 13

1. $K_0(k, n) = 1, K_1(k, n) = n - 2k$,
2. $(i+1)K_{i+1}(k, n) = (n-2k)K_i(k, n) - (n-i+1)K_{i-1}(k, n)$,
3. $K_i(k, n) = (-1)^k K_{n-i}(k, n)$ *(This implies, for n even and k odd, $K_{\frac{n}{2}}(k, n) = 0$)*,
4. $\binom{n}{k} K_i(k, n) = \binom{n}{i} K_k(i, n)$,
5. $K_i(k, n) = (-1)^i K_i(n-k, n)$, *(This implies, for n even and i odd, $K_i(\frac{n}{2}, n) = 0$)*,
6. $(n-k)K_i(k+1, n) = (n-2i)K_i(k, n) - kK_i(k-1, n)$,
7. $(n-i+1)K_i(k, n+1) = (3n-2i-2k+1)K_i(k, n) - 2(n-k)K_i(k, n-1)$.

Proposition 14 *For n even,* $K_i(\frac{n}{2}, n) = \begin{cases} 0 \text{ for odd } i. \\ (-1)^{\frac{i}{2}} \binom{\frac{n}{2}}{\frac{i}{2}} \text{ for even } i. \end{cases}$

Proof : For odd i, it is proved in Proposition 13(Item 5). Now we will prove for even i using induction on i. For the base step, i.e., $i = 0$, we have $K_0(\frac{n}{2}, n) = \binom{\frac{n}{2}}{0} = 1$. We will prove inductive step. Suppose it is true for $i = l$, i.e., $K_l(\frac{n}{2}, n) = (-1)^{\frac{l}{2}} \binom{\frac{n}{2}}{\frac{l}{2}}$. Now we will prove for

$i = l+2$. Following Proposition 13(Item 2), we have $(l+2)K_{l+2}(\frac{n}{2}, n) = -(n-l)K_l(\frac{n}{2}, n)$ (in the proposition, we put $l+1$ instead of i). So, $K_{l+2}(\frac{n}{2}, n) = (-1)^{\frac{l}{2}+1} \frac{n-l}{l+2}\binom{\frac{n}{2}}{\frac{l}{2}} = (-1)^{\frac{l}{2}+1}\binom{\frac{n}{2}}{\frac{l}{2}+1}$. Hence proved. ■

Let us now concentrate on the Walsh spectra of the symmetric function ψ_n as explained in Construction 6.

Lemma 7 *Consider the function ψ_n on n number of variables as given in Construction 6.*

1. *For k even,* $rw_{\psi_n}(k) = \begin{cases} K_{\frac{n}{2}}(k, n) \text{ for even } n. \\ 0 \text{ for odd } n. \end{cases}$
2. *For k odd,* $rw_{\psi_n}(k) = 2 \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} K_i(k, n).$
3. $rw_{\psi_n}(1) = 2\binom{n-1}{\lfloor \frac{n}{2} \rfloor}.$
4. $rw_{\psi_n}(n) = \begin{cases} (-1)^{\frac{n}{2}}\binom{n}{\frac{n}{2}} \text{ for even } n. \\ (-1)^{\frac{n-1}{2}} 2\binom{n-1}{\frac{n-1}{2}} \text{ for odd } n. \end{cases}$
5. *For even n,* $rw_{\psi_n}(\frac{n}{2}) = \begin{cases} (-1)^{\frac{n}{4}}\binom{\frac{n}{2}}{\frac{n}{4}} \text{ for even } \frac{n}{2}. \\ 2\sum_{i=0}^{\frac{n-2}{4}} (-1)^i \binom{\frac{n}{2}}{i} \text{ for odd } \frac{n}{2}. \end{cases}$

Proof : From Equation 5.4 we have

$$rw_{\psi_n}(k) = \sum_{i=0}^{n} (-1)^{re_{\psi_n}(i)} K_i(k, n) = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} K_i(k, n) - \sum_{i=\lfloor \frac{n}{2} \rfloor + 1}^{n} K_i(k, n),$$

as

$$re_{\psi_n}(i) = \begin{cases} 0 \text{ for } 0 \leq i \leq \lfloor \frac{n}{2} \rfloor \\ 1 \text{ for } \lfloor \frac{n}{2} \rfloor < i \leq n. \end{cases}$$

Moreover, from Proposition 13(Item 3), we have $K_i(k,n) = (-1)^k K_{n-i}(k,n)$, i.e., if k is even, $K_i(k,n) = K_{n-i}(k,n)$. Now, $\sum_{i=\lfloor \frac{n}{2} \rfloor + 1}^{n} K_i(k,n) = \sum_{j=0}^{\lceil \frac{n}{2} \rceil - 1} K_{j+\lfloor \frac{n}{2} \rfloor + 1}(k,n) = \sum_{j=0}^{\lceil \frac{n}{2} \rceil - 1} K_{n-j}(k,n) = \sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1} K_i(k,n) = \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} K_i(k,n)$. Hence, $rw_{\psi_n}(k) = K_{\frac{n}{2}}(k,n)$ for even n and $rw_{\psi_n}(k) = 0$ for odd n. This proves the first item.

Here we will prove Item 2. From Proposition 13(Item 3), we have $K_i(k,n) = -K_{n-i}(k,n)$ as k is odd. Following the line of the proof of Item 1, we get $rw_{\psi_n}(k) = 2\sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} K_i(k,n)$ (the even n and odd k case is handled under the same formula as $K_{\frac{n}{2}(k,n)} = 0$). So, we prove the second item.

To prove Item 3, we have from Equation 5.3 that $K_i(1,n) = \binom{n-1}{i} - \binom{n-1}{i-1}$. Thus, following item 2, $rw_{\psi_n}(1) = 2\sum_{i=0}^{\lceil \frac{n}{2} \rceil - 1}(\binom{n-1}{i} - \binom{n-1}{i-1}) = 2\binom{n-1}{\lceil \frac{n}{2} \rceil - 1}$. So, for odd n, $rw_{\psi_n}(1) = 2\binom{n-1}{\frac{n-1}{2}}$ and for even n, $rw_{\psi_n}(1) = 2\binom{n-1}{\frac{n}{2}-1} = 2\binom{n-1}{\frac{n}{2}}$. Therefore for any n, $rw_{\psi_n}(1) = 2\binom{n-1}{\lfloor \frac{n}{2} \rfloor}$.

For the fourth item, note that, $K_i(n,n) = (-1)^i K_i(0,n) = (-1)^i \binom{n}{i}$. For n even, following item 1, $rw_{\psi_n}(n) = K_{\frac{n}{2}}(n,n) = (-1)^{\frac{n}{2}} K_{\frac{n}{2}}(0,n) = (-1)^{\frac{n}{2}} \binom{n}{\frac{n}{2}}$. For odd n, following item 2, $rw_{\psi_n}(n) = 2\sum_{i=0}^{\frac{n-1}{2}} (-1)^i \binom{n}{i} = 2\sum_{i=0}^{\frac{n-1}{2}} (-1)^i (\binom{n-1}{i} + \binom{n-1}{i-1}) = \pm 2\binom{n-1}{\frac{n-1}{2}}$ (positive when $n = 1 \bmod 4$, negative when $n = 3 \bmod 4$).

For fifth item, following Item 1 of this lemma and Proposition 14, the case for $\frac{n}{2}$ even is proved. Similarly, following Item 2 of this lemma and Proposition 14, the case for $\frac{n}{2}$ odd is proved. ■

Lemma 8 *For $1 \leq k \leq \lfloor \frac{n-1}{2} \rfloor$ and $0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor$, $K_i(1,n) \geq |K_i(k,n)|$.*

Proof : Note that, $K_i(1,n) = \binom{n-1}{i} - \binom{n-1}{i-1} \geq 0$ for $0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor$ and that implies $|K_i(1,n)| = K_i(1,n)$ in $0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor$.

First, we will prove it for $i \geq k$ using induction on k. In this direction for the base step we need to show $K_i(1,n) \geq |K_i(1,n)|$ (which is obvious) and $K_i(1,n) \geq |K_i(2,n)|$. Now $K_i(2,n) = \binom{n-2}{i} - 2\binom{n-2}{i-1} + \binom{n-2}{i-2}$ and $K_i(1,n) = \binom{n-1}{i} - \binom{n-1}{i-1} = \binom{n-2}{i} + \binom{n-2}{i-1} - \binom{n-2}{i-1} - \binom{n-2}{i-2} = \binom{n-2}{i} - \binom{n-2}{i-2}$. If $K_i(2,n) \geq 0$ then $K_i(1,n) - K_i(2,n) = 2(\binom{n-2}{i-1} - \binom{n-2}{i-2}) \geq 0$ as $(i-1) \leq \lfloor \frac{n-2}{2} \rfloor$. If $K_i(2,n) \leq 0$ then $K_i(1,n) + K_i(2,n) = 2(\binom{n-2}{i} - \binom{n-2}{i-1}) \geq 0$ for $i \leq \lfloor \frac{n-2}{2} \rfloor$.

Note that, $\lfloor \frac{n-1}{2} \rfloor = \lfloor \frac{n-2}{2} \rfloor$ when n is even and $\binom{n-2}{i} - \binom{n-2}{i-1} = 0$ for $i = \lfloor \frac{n-1}{2} \rfloor$ when n is odd. Therefore, $|K_i(1,n)| \geq |K_i(2,n)|$, i.e., $K_i(1,n) \geq |K_i(2,n)|$. Thus the base steps are proved.

Suppose for some $1 \leq k < \lfloor \frac{n-1}{2} \rfloor$, $K_i(1,n) \geq |K_i(j,n)|$ for all j, $1 \leq j \leq k$. Now we will prove $K_i(1,n) \geq |K_i(k+1,n)|$. From Proposition 13(6), we have
$(n-k)K_i(k+1,n) = (n-2i)K_i(k,n) - kK_i(k-1,n)$,
i.e., $(n-k)|K_i(k+1,n)| \leq (n-2i)|K_i(k,n)| + k|K_i(k-1,n)|$,
i.e., $(n-k)|K_i(k+1,n)| \leq (n-2i)K_i(1,n) + kK_i(1,n)$,
i.e., $|K_i(k+1,n)| \leq \frac{n-2i+k}{n-k} K_i(1,n)$,
i.e., $|K_i(k+1,n)| \leq K_i(1,n)$, since $\frac{n-2i+k}{n-k} \leq 1$ for $i \geq k$. So, the proof is completed for $j = k+1$. Hence, $K_i(1,n) \geq |K_i(k,n)|$ for $0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor$, $1 \leq k \leq \lfloor \frac{n-1}{2} \rfloor$ and $i \geq k$.

Now we will prove for $0 \leq i < k \leq \lfloor \frac{n-1}{2} \rfloor$. Since $k > i$, following the above proof, we have $K_k(1,n) \geq |K_k(i,n)|$ by interchanging the role of k and i. Thus, $\binom{n}{i} K_k(1,n) \geq \binom{n}{i} |K_k(i,n)|$. Now following Proposition 13(4), we have $\binom{n}{i} K_k(1,n) \geq \binom{n}{k} |K_i(k,n)|$, i.e.,

$$\frac{\binom{n}{i}}{\binom{n}{k}} K_k(1,n) \geq |K_i(k,n)|. \tag{5.5}$$

Further, following Proposition 13(4), we have $K_k(1,n) = \frac{\binom{n}{k}}{\binom{n}{1}} K_1(k,n) = \frac{\binom{n}{k}}{n}(n-2k)$ and $K_i(1,n) = \frac{\binom{n}{i}}{n}(n-2i)$. So, $\frac{K_k(1,n)}{K_i(1,n)} = \frac{\binom{n}{k}(n-2k)}{\binom{n}{i}(n-2i)}$, i.e., $K_k(1,n) = \frac{\binom{n}{k}(n-2k)}{\binom{n}{i}(n-2i)} K_i(1,n)$. Now putting the value of $K_k(1,n)$ in Equation 5.5, we have $\frac{n-2k}{n-2i} K_i(1,n) \geq |K_i(k,n)|$, i.e., $K_i(1,n) \geq |K_i(k,n)|$, since $\frac{n-2k}{n-2i} < 1$ as $i < k$. Hence the proof. ∎

In the next corollary we extend the range of i and k.

Corollary 11

1. *For odd n, $|K_i(1,n)| \geq |K_i(k,n)|$ where $0 \leq i \leq n$ and $1 \leq k \leq n-1$.*
2. *For even n, $|K_i(1,n)| \geq |K_i(k,n)|$ where $0 \leq i \leq n$ and $1 \leq k \leq n-1$ except $i = \frac{n}{2}$ or $k = \frac{n}{2}$.*

Proof : The proof for $0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor$ and $1 \leq k \leq \lfloor \frac{n-1}{2} \rfloor$ is done in Lemma 8. The remaining part can be proved using the symmetry relations $K_i(k,n) = (-1)^k K_{n-i}(k,n)$ and $K_i(k,n) = (-1)^i K_i(n-k,n)$ in Proposition 13(Item 3 and Item 5). ∎

When n is even the relation proved above is not true for $i = \frac{n}{2}$ and even k, since $K_{\frac{n}{2}}(1,n) = 0$ and $K_{\frac{n}{2}}(k,n)$ is a non zero number for even k.

Theorem 13 *Consider the functions $\psi_n \in \mathcal{B}_n$, as explained in Construction 6. Then* $\mathsf{nl}(\psi_n) = 2^{n-1} - \binom{n-1}{\lfloor \frac{n}{2} \rfloor}$.

Proof : First we prove that $rw_{\psi_n}(1)$ is maximum among all $rw_{\psi_n}(k)$ in $0 \leq k \leq n$.

Case 1. Let n be odd. First we show that $|rw_{\psi_n}(k)| \leq rw_{\psi_n}(1)$ for all k in the range $1 \leq k \leq n-1$. We know, $|rw_{\psi_n}(k)| = |2\sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} K_i(k,n)| \leq 2\sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} |K_i(k,n)|$. From Lemma 8 we have, $K_i(1,n) \geq |K_i(k,n)|$ for $1 \leq k \leq n-1$, and $0 \leq i \leq \lfloor \frac{n-1}{2} \rfloor$. This gives, $|rw_{\psi_n}(k)| \leq 2\sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} K_i(1,n) = rw_{\psi_n}(1)$. Again from Lemma 7 we have, $rw_{\psi_n}(1) = |rw_{\psi_n}(n)|$. Finally $rw_{\psi_n}(0) = 0$. Hence $rw_{\psi_n}(1) \geq |rw_{\psi_n}(k)|$ for $0 \leq k \leq n$.

Case 2. Let n be even. Let us first consider that k is odd and in $1 \leq k \leq n-1$ except $k = \frac{n}{2}$. From Lemma 7 we get that $|rw_{\psi_n}(k)| = |2\sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} K_i(k,n)|$. So following the same argument used in the previous case, we get $|rw_{\psi_n}(k)| \leq rw_{\psi_n}(1)$. For $k = \frac{n}{2}$ odd, from Lemma 7(item 5) we have $|rw_{\psi_n}(\frac{n}{2})| = |2\sum_{i=0}^{\frac{n-2}{4}} (-1)^i \binom{\frac{n}{2}}{i}| \leq 2\sum_{i=0}^{\frac{n-2}{4}} \binom{\frac{n}{2}}{i} = 2^{\frac{n}{2}}$. By induction on n it can be proved that $2^{\frac{n}{2}} \leq 2\binom{n-1}{\frac{n}{2}} = rw_{\psi_n}(1)$. So, for k odd and $1 \leq k \leq n-1$, the proof is done. When k even and $2 \leq k \leq n-2$, we have from Lemma 7 that $rw_{\psi_n}(k) = K_{\frac{n}{2}}(k,n)$. Now $K_{\frac{n}{2}}(k,n) = \sum_{j=0}^{\frac{n}{2}} (-1)^j \binom{k}{j}\binom{n-k}{\frac{n}{2}-j} \leq \sum_{j=0}^{\frac{n}{2}} \binom{k}{j}\binom{n-k}{\frac{n}{2}-j} = \binom{n}{\frac{n}{2}} = rw_{\psi_n}(1)$. Further, since $K_{\frac{n}{2}}(0,n) = \binom{n}{\frac{n}{2}} = |K_{\frac{n}{2}}(n,n)|$, we get, $rw_{\psi_n}(1) = rw_{\psi_n}(0) = |rw_{\psi_n}(n)|$. Thus $|rw_{\psi_n}(k)| \leq rw_{\psi_n}(1)$ for all k in $0 \leq k \leq n$.

So for any n, $\mathsf{nl}(\psi_n) = 2^{n-1} - \frac{1}{2}|rw_{\psi_n}(1)| = 2^{n-1} - \binom{n-1}{\lfloor \frac{n}{2} \rfloor}$. ∎

Like the proof of nonlinearity of ϕ_{2k} in Proposition 8 (Item 3) in Chapter 4, using the result in Theorem 4 in Chapter 3, one can proof the nonlinearity of ψ_{2k} in a simpler way as follows. According the last paragraph of Section 6.1 in Chapter 6 we have $\mathsf{AI}_{2k+1}(x_{2k+1} + \psi_{2k}) = k+1$. Hence, by Theorem 4, we have $\mathsf{nl}(x_{2k+1} + \psi_{2k}) \geq 2^{2k} - \binom{2k}{k}$ which implies $\mathsf{nl}(\psi_{2k}) \geq 2^{2k-1} - \frac{1}{2}\binom{2k}{k}$. On the other hand, since the $\mathsf{wt}(\psi_{2k}) = 2^{2k-1} - \frac{1}{2}\binom{2k}{k}$, $\mathsf{nl}(\psi_{2k}) \leq 2^{2k-1} - \frac{1}{2}\binom{2k}{k}$. Therefore the $\mathsf{nl}(\psi_{2k}) = 2^{2k-1} - \frac{1}{2}\binom{2k}{k} = 2^{2k-1} - \binom{2k-1}{k}$.

Though this proof is simpler than our original proof, our proof was the first available proof for the result. Further, by this study, one can understand the structure of the function ψ_n which are related to Krawtchouk polynomials. Further the results related to Krawtchouk polynomial can be used to find out the Walsh Spectra values of ψ_n at all the points.

Little later in [22], Braeken et. al. independently presented that function ψ_n has optimal AI with a different proof technique. They have also studied the nonlinearity of these functions, but only experimentally.

Now we would like to present a few observations.

1. We have checked for odd n up to $n = 11$, the function we have constructed in Construction 6, is the only function with maximum possible algebraic immunity among the symmetric functions. There is no other symmetric Boolean function on odd number of variables that are of algebraic immunity $\lceil \frac{n}{2} \rceil$ as far as we have experimented. This is an important open question to be proved or disproved.

2. For even n, we have found that there are symmetric functions with full algebraic immunity other than what we have presented in Construction 6. In fact so far we have experimented, up to $n = 12$, we found functions with full algebraic immunity $\frac{n}{2}$ and nonlinearity greater than that of the function constructed in Construction 6. In Table 5.1, we present the maximum nonlinearity available for symmetric Boolean functions on even number of variables having maximum possible algebraic immunity. This we found by computer search by writing computer program. In [22], some more classes of symmetric Boolean functions having optimal AI have been characterized.

n	4	6	8	10	12
nonlinearity of Construction 6	5	22	93	386	1586
maximum nonlinearity (by exhaustive search)	6	26	94	394	1630

Table 5.1: Nonlinearity of symmetric Boolean functions on even number of variables by Construction 6 and maximum nonlinearity by exhaustive search.

5.3 Results Comparing that of ϕ_{2k} in Chapter 4

We have proved in Theorem 13 that the nonlinearity of the functions $\psi_{2k},\ k > 0$ in Construction 6 are same as their weight (i.e., both $2^{n-1} - \binom{n-1}{\frac{n}{2}}$). Interestingly their weight and nonlinearity are equal with that for the function ϕ_{2k} in Chapter 4. This because the lowest degree of annihilators of both the functions ϕ_{2k} and ψ_{2k} are at degree k and the lowest degree annihilators of their complements are at degree $k + 1$ (see Corollary 7 of Chapter 4 and Corollary 9 of Chapter 5), which implies that their nonlinearity and weight same to each other (see Proposition 8 of Chapter 4 and nonlinearity calculation using Lobanov result [111] in the previous section). However, these functions are not same and further not always affine transformation of ϕ_{2k} as the algebraic degree of the functions are different from that of ϕ_{2k} as presented in Table 5.2.

$n = 2k$	2	4	6	8	10	12	14	16
$\deg(\phi_{2k})$	2	4	5	8	9	11	13	16
$\deg(\psi_{2k})$	2	4	4	8	8	8	8	16

Table 5.2: Comparison of algebraic degree.

5.4 Construction of Balanced Functions

Let us now concentrate on construction of balanced f with maximum possible algebraic immunity for even n. Refer to the general form of f as given in Construction 5. If b is so chosen that out of $\binom{n}{\frac{n}{2}}$ inputs, half of the corresponding outputs are 1 and the other half are 0, then f will be balanced. To formalize it, consider two sets $S_n, T_n \subset \{x \in \mathbb{F}_2^n \mid \mathsf{wt}(x) = \frac{n}{2}\}$, such that $S_n \cap T_n = \emptyset$ and $|S_n| = |T_n| = \frac{1}{2}\binom{n}{\frac{n}{2}}$. Note that there are $\binom{\binom{n}{\frac{n}{2}}}{\frac{1}{2}\binom{n}{\frac{n}{2}}} = \binom{\binom{n}{\frac{n}{2}}}{\binom{n-1}{\frac{n}{2}}}$ many different options to choose any S_n and correspondingly a T_n.

Now we have the following result.

Proposition 15 *Let F be an n-variable balanced function (n even) as follows.*

$$\begin{aligned} F(x_1, \ldots, x_n) &= 0 \text{ for } wt(x_1, \ldots, x_n) < \frac{n}{2}, \\ &= 1 \text{ for } wt(x_1, \ldots, x_n) > \frac{n}{2}, \\ &= 0 \text{ for } (x_1, \ldots, x_n) \in S_n, \\ &= 1 \text{ for } (x_1, \ldots, x_n) \in T_n. \end{aligned}$$

Then $\mathsf{nl}(F) \geq 2^{n-1} - \binom{n}{\frac{n}{2}}$.

Proof : Consider the function f in Construction 6. It is clear that $\frac{1}{2}\binom{n}{\frac{n}{2}}$ many output points in the truth table of f need to be toggled to get the function F. Thus $\mathsf{nl}(F) \geq \mathsf{nl}(f) - \frac{1}{2}\binom{n}{\frac{n}{2}}$. From Theorem 13, $\mathsf{nl}(f) = 2^{n-1} - \binom{n-1}{\frac{n}{2}}$. Thus $\mathsf{nl}(F) \geq 2^{n-1} - \binom{n-1}{\frac{n}{2}} - \frac{1}{2}\binom{n}{\frac{n}{2}} = 2^{n-1} - \frac{1}{2}\binom{n}{\frac{n}{2}} - \frac{1}{2}\binom{n}{\frac{n}{2}} = 2^{n-1} - \binom{n}{\frac{n}{2}}$. ∎

However, we now show a heuristic construction with which we can actually get much better value of nonlinearity of the balanced functions. Note that we do not present any theoretical proof here, but only list the experimental results.

For that we first refer to Maiorana-McFarland type of bent functions. The Maiorana-McFarland class of bent function is as follows [72]. Consider n-variable Boolean functions on

(X, Y), where $X, Y \in \mathbb{F}_2^{\frac{n}{2}}$ of the form $f(X, Y) = X \cdot \pi(Y) + g(Y)$ where π is a permutation on $\mathbb{F}_2^{\frac{n}{2}}$ and g is any Boolean function on $\frac{n}{2}$ variables. The function f can be seen as concatenation of $2^{\frac{n}{2}}$ distinct (up to complementation) affine function on $\frac{n}{2}$ variables. For our purpose we consider π as an identity permutation, g as a constant zero function and refer to this function on n variables as $b(x_1, \ldots, x_n)$, for n even. Now we construct an n-variable function G as follows.

$$\begin{aligned} G(x_1, \ldots, x_n) &= 0 \text{ for } wt(x_1, \ldots, x_n) < \frac{n}{2}, \\ &= 1 \text{ for } wt(x_1, \ldots, x_n) > \frac{n}{2}, \\ &= b(x_1, \ldots, x_n) \text{ for } wt(x_1, \ldots, x_n) = \frac{n}{2}. \end{aligned}$$

Experimentally we observe that $\mathsf{nl}(G) = \mathsf{nl}(f)$, for even n up to 16, where f is the function as described in Construction 6. Note that G is much closer to balancedness than the function f.

1. If $\mathsf{wt}(G) < 2^{n-1}$, then we choose $2^{n-1} - \mathsf{wt}(G)$ points randomly from the inputs having weight $\frac{n}{2}$ and output 0 of G and toggle those outputs to 1.
2. If $\mathsf{wt}(G) > 2^{n-1}$, then we choose $\mathsf{wt}(G) - 2^{n-1}$ points randomly from the inputs having weight $\frac{n}{2}$ and output 1 of G and toggle those outputs to 0.

After this change G will become balanced. Experimentally we get the following result for the function G in Table 5.3. We execute 100 runs for each n and take the best result among the runs in terms of nonlinearity. We also observe that algebraic degree of the reported functions is the maximum possible, i.e., $n - 1$.

$n = 2k$	4	6	8	10	12	14	16
$2^{n-1} - \binom{n-1}{\frac{n}{2}}$	5	22	93	386	1586	6476	26333
$\mathsf{nl}(G)$	4	22	92	384	1582	6468	26316
$4(2^{n-3} - \binom{n-3}{\frac{n-2}{2}})$	4	20	88	372	1544	6344	25904

Table 5.3: Comparison of nonlinearities.

We have also checked that the degree of G is always the maximum possible, i.e., $n - 1$, for a balanced function.

As by itself the function ϕ_{2k} was not balanced, the construction of balanced function (using the strategy of Chapter 4 with full algebraic immunity is basically $x_1 + x_2 + \phi_{2k-2}$, where ϕ_{2k-2} was on the variables $x_3, \ldots, x_{2k}$. The nonlinearity of this function is $4\,\mathsf{nl}(\phi_{2k-2}) = 4(2^{n-3} - \binom{n-3}{\frac{n-2}{2}})$. That is presented in the last row of Table 5.3. Clearly this heuristic construction presents better nonlinearity than this.

5.5 Conclusion

In this chapter we have presented the basic theory towards the construction of Boolean functions with full algebraic immunity. Based on this theory we present some concrete construction ideas. Further we could study the other cryptographic properties like nonlinearity and algebraic degree in detail.

Chapter 6

Resistance of Boolean Functions against Fast Algebraic Attack: Study and Construction

Algebraic and fast algebraic attacks have recently received a lot of attention in cryptographic literature [3, 7, 43, 58, 56, 51, 109, 123]. Using good algebraic immunity one may achieve resistance against algebraic attacks done in a particular way, i.e., using annihilators and linearization. In fact, one may not need linearization if algorithms using Gröbner bases can be properly exploited. Further it should be noted that based on some recent works related to fast algebraic attacks [5, 51, 20, 4], one should concentrate more carefully on the design parameters of Boolean functions for proper resistance. The weakness of algebraic immunity against fast algebraic attack has been demonstrated in [53] by mounting an attack on SFINKS [19]. We have discussed more details in Section 2.2.3 of Chapter 2.

Consider a function f with maximum possible algebraic immunity $\lceil \frac{n}{2} \rceil$. It may very well happen that in that case $f * g = h$, where $\deg(h) = \lceil \frac{n}{2} \rceil$, but $\deg(g) < \lceil \frac{n}{2} \rceil$. Subsequently the lower degree of g may be exploited to mount a fast attack (well known as fast algebraic attack) even if the algebraic immunity of f is the maximum possible. In fact, there are examples, where one can get a linear g too. Initial study of Boolean functions in this area has been started in [20, 4]. Since algebraic immunity is now understood as a necessary (but not sufficient) condition against resisting algebraic and fast algebraic attacks, we feel there is a need to consider the functions with full algebraic immunity for their performance in terms of $f * g = h$ relationship. That is for the functions f with full algebraic immunity we consider $\deg(h) \geq \lceil \frac{n}{2} \rceil$, and then after fixing the degree of h, we try to get the minimum degree g.

One should be aware that checking these $f * g = h$ relationships are not all and there are number of scenarios to mount algebraic and fast algebraic attacks which are available in details in [56, 51].

It is always meaningful to consider $f * g = h$ only when $\deg(g) \leq \deg(h)$ as otherwise $f * g = h$ will imply $f * h = h$. So for all the discussion in this chapter we will consider $\deg(g) \leq \deg(h)$ for a relation $f * g = h$ unless mentioned otherwise.

In this chapter, we present a specific class of balanced functions f for even number of input variables n having algebraic immunity $\frac{n}{2}$ such that for any $f * g = h$ relation if $\deg(h) = \frac{n}{2}$ then $\deg(g)$ cannot be less than $\frac{n}{2}$. This class of functions was not known earlier. Further we show that existence of these functions has direct implication towards existence of resilient functions with maximum possible algebraic immunity.

6.1 Algebraic Immunity of f and the $f * g = h$ Relationships

In this section we present some basic results.

Proposition 16 *Consider an n-variable (n odd) function f having $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$. Then*

1. *there will always exist g, h, such that $f * g = h$, where $\deg(g) = \lfloor \frac{n}{2} \rfloor$ and $\deg(h) = \lceil \frac{n}{2} \rceil$.*
2. *there never exist g, h, such that $f * g = h$, where $\deg(g) = \lfloor \frac{n}{2} \rfloor$ and $\deg(h) < \lceil \frac{n}{2} \rceil$.*

Proof : By [51, Theorem 7.2.1], we know that there always exists g, h, such that $f * g = h$, with $\deg(g) + \deg(h) = n$. Thus, if we fix $\deg(g) = \lfloor \frac{n}{2} \rfloor$ and $\deg(h) = \lceil \frac{n}{2} \rceil$, we get the required result for first item. The second item comes from the fact that $\mathsf{AI}_n(f) = \lceil \frac{n}{2} \rceil$. ■

This always means that even if a function on odd number of variables n has full algebraic immunity $\lceil \frac{n}{2} \rceil$, one will always get a g one degree lower than that. However, for even n, this may or may not be true. Later in this chapter we will show that given a Boolean function on n variables with full algebraic immunity $\frac{n}{2}$, one may or may not get a g having degree less than $\frac{n}{2}$ such that $f * g = h$ when $\deg(h) = \frac{n}{2}$.

Proposition 17 *Consider an n-variable function f. Consider the relationship $f * g = h$, such that $\deg(h) = \mathsf{AI}_n(f)$. Then if $\deg(g) < \mathsf{AI}_n(f)$ then both f and $1 + f$ have minimum degree annihilators at degree $\mathsf{AI}_n(f)$.*

Proof : It is clear that at least one of f and $1+f$ will have an annihilator at degree $\mathsf{AI}_n(f)$. Without loss of generality, consider that f has the minimum degree annihilator at degree $\mathsf{AI}_n(f)$ and $1+f$ has the minimum degree annihilator at degree greater than or equal to $\mathsf{AI}_n(f)$. Consider the relations of the form $f*g=h$, when $\deg(g)<\deg(h)$ and $\deg(h)=\mathsf{AI}_n(f)$. From [20], $f*g=h$ iff $f*(g+h)=0$ and $(1+f)*h=0$. As $\deg(g)<\deg(h)$, we have $\deg(g+h)=\deg(h)=\mathsf{AI}_n(f)$. Thus $1+f$ has an annihilator at degree $\mathsf{AI}_n(f)$. ■

The following corollary is immediate from Proposition 17.

Corollary 12 *Let only one of f and $1+f$ has minimum degree annihilator at $\mathsf{AI}_n(f)$ and the other one has minimum degree annihilator at degree greater than $\mathsf{AI}_n(f)$. Then there is no $f*g=h$ relation having $\deg(h)=\mathsf{AI}_n(f)$ and $\deg(g)<\mathsf{AI}_n(f)$.*

We also present the following result that can be used to find minimum degree g in the relation $f*g=h$, where $\deg(h)=\mathsf{AI}_n(f)$.

Proposition 18 *Consider that $f, 1+f$ have minimum degree annihilators at the same degree $\mathsf{AI}_n(f)$. Let A be the set of annihilators of f and B be the set of annihilators of $1+f$ at degree $\mathsf{AI}_n(f)$. Then the minimum degree of g such that $f*g=h$ is $\min\limits_{\beta_A\in A,\beta_B\in B}\deg(\beta_A+\beta_B)$, where h is a function of degree $\mathsf{AI}_n(f)$.*

Also we present the following result relating g and h only.

Proposition 19 *If $f*g=h$, then $g*h=h$, i.e., h is the annihilator of $1+g$.*

Proof : We have, $f*g=h$, i.e., $f*g*g=g*h$, i.e., $f*g=g*h$, i.e., $h=g*h$. ■

Consider two functions $\tau_1,\tau_2\in\mathcal{B}_n$ having full algebraic immunity $\lceil\frac{n}{2}\rceil$ when n is odd. If we consider the function $\tau=(1+x_{n+1})\tau_1+x_{n+1}\tau_2$, on even number of variables, it can be checked using Proposition 3(2) in Chapter 3 that this is again of full algebraic immunity $\frac{n+1}{2}$ which is actually $\lceil\frac{n}{2}\rceil$.

However, the situation is not as simple when we take n even. In such a situation we start with two functions $\tau_1,\tau_2\in\mathcal{B}_n$ having full algebraic immunity $\frac{n}{2}$. In that case, $\tau=(1+x_{n+1})\tau_1+x_{n+1}\tau_2$, on odd number of variables may or may not have full algebraic immunity $\lceil\frac{n+1}{2}\rceil=\frac{n}{2}+1$.

Consider τ_1,τ_2 have annihilators π_1,π_2 at degree $\frac{n}{2}$ and $1+\tau_1, 1+\tau_2$ have annihilators π_1',π_2' at degree $\frac{n}{2}$. Then following the Proposition 3(2) in Chapter 3, τ will have algebraic immunity $\frac{n}{2}$, iff $\deg(\pi_1+\pi_2)<\frac{n}{2}$ or $\deg(\pi_1'+\pi_2')<\frac{n}{2}$.

Now consider that τ_1, τ_2 have minimum degree annihilators π_1, π_2 at degree $\frac{n}{2}$ and $\frac{n}{2}+1$ respectively. Further $1+\tau_1, 1+\tau_2$ have minimum degree annihilators π_1', π_2' at degree $\frac{n}{2}+1$ and $\frac{n}{2}$ respectively. Then one can check that τ has algebraic immunity $\frac{n}{2}+1$. Note that the functions ϕ_{2k} (see Section 6.2) and the functions ψ_{2k} (see Section 6.3) have the properties like τ_1 and $1+\phi_{2k}$, $1+\psi_{2k}$ have the properties like τ_2. Thus the availability of the functions ϕ_{2k}, ψ_{2k} having full algebraic immunity k presents a clear construction using them to get functions with full algebraic immunity $k+1$ on odd number of variables $2k+1$. As concrete examples, $x_{2k+1}+\phi_{2k}$, $x_{2k+1}+\psi_{2k}$, $(1+x_{n+1})\phi_{2k}+x_{n+1}(1+\psi_{2k})$, $(1+x_{n+1})\psi_{2k}+x_{n+1}(1+\phi_{2k})$ are functions on odd number of variables with full algebraic immunity.

6.2 Study of ϕ_{2k} from Chapter 4

In Corollary 7 of Chapter 4, it is proved that the minimum degree annihilators of ϕ_{2k} are at the degree k and the the minimum degree annihilators of $1+\phi_{2k}$ are at the degree $k+1$. Then using Corollary 12, we get that there is no g having degree less than k such that $\phi_{2k}g = h$, where $\deg(h) = k$.

Theorem 14 *Let $f \in \mathcal{B}_{2k}$ such that the degree of minimum degree annihilators of f and $1+f$ are d and e respectively, $d, e > 0$. Suppose there exist $g, h \in \mathcal{B}_{2k}$ such that $f * g = h$, where g is a non zero function. Then either h is zero or $\deg(h) \geq e$. If h is zero then $\deg(g) \geq d$.*

Proof : If possible, consider that there exists a nonzero h of degree $e_1 < e$. Then from the result [20, Lemma 1] that $f * g = h$ iff $f * (g+h) = 0$ and $(1+f) * h = 0$, we find h is an e_1 degree annihilator of $1+f$ which is a contradiction. Further if h is a zero function then $f * g = 0$. As f has no annihilator of degree less than d and g is a non zero function, $\deg(g) \geq d$. ■

Now consider any function $f \in \mathcal{B}_{2k}$ such that the minimum degree annihilators of f and $1+f$ are at degree k and $k+1$. Then following Theorem 14, we can not find any such nonzero h of degree less than $k+1$. If we take h as a zero function then degree of g has to be greater than or equal to k. Since ϕ_{2k} has minimum degree annihilator at degree k and $1+\phi_{2k}$ has minimum degree annihilator at degree $k+1$, we get the following result.

Corollary 13 *Consider $g, h \in \mathcal{B}_{2k}$ such that $\phi_{2k}g = h$ where $g \neq 0$. Then either $\deg(h) > k$ or if $h = 0$ then $\deg(g) \geq k$.*

$2k$	$\deg(g)$	$\deg(h)$
6	1	4
8	1	5

$2k$	$\deg(g)$	$\deg(h)$
12	3	7
12	3	8
12	1	9

$2k$	$\deg(g)$	$\deg(h)$
10	2	6
10	2	7
10	1	8

$2k$	$\deg(g)$	$\deg(h)$
14	4	8
14	4	9
14	2	10
14	2	11
14	1	12

Table 6.1: Experimental results on $\phi_{2k}g = h$ relationship.

Note that this means one cannot get a lower degree (than $\mathsf{AI}_{2k}(\phi_{2k}) = k$) function g by fixing h at a degree k. Note that in [4, Table 3], the functions on $2k$ variables are not ϕ_{2k}, but the functions of the form $x_1x_2 + \phi_{2k-2}(x_3, \ldots, x_{2k})$ which are also of full algebraic immunity k. That is why those functions are weak against fast algebraic attack. Further in case of $\deg(h) > k$, we present the experimental results in Table 6.1 for the $\phi_{2k}g = h$ relationships for $6 \leq 2k \leq 14$. We present the minimum degree of g in the table till it becomes 1.

From Table 6.1, it is clear that with the increase of $\deg(h)$, the degree of g decreases as expected, but the rate is not sharp. In fact, if one uses ϕ_{14}, then one gets a linear g only when h is of degree 12. Thus we like to point out that though the function ϕ_{2k} is not good in terms of nonlinearity, its structure is good for immunity against both algebraic and fast algebraic attacks.

6.3 Study on Symmetric Functions

Construction for symmetric functions ψ_{2k} with maximum algebraic immunity has been presented in Construction 6 in Chapter 3. $\psi_n \in \mathcal{B}_n$, as follows:

$$\psi_n(x) = \begin{cases} 0 \text{ for } \mathsf{wt}(x) \leq \lceil \frac{n}{2} \rceil, \\ 1 \text{ for } \mathsf{wt}(x) > \lceil \frac{n}{2} \rceil. \end{cases}$$

We have from Corollary 9 in Chapter 5 that ψ_{2k} has minimum degree annihilators at degree k and $1 + \psi_{2k}$ has minimum degree annihilators at degree $k + 1$. Thus, similar to

$2k$	$\deg(g)$	$\deg(h)$
6	0	4
8	1	5

$2k$	$\deg(g)$	$\deg(h)$
10	2	6
10	2	7
10	0	8

$2k$	$\deg(g)$	$\deg(h)$
12	3	7
12	0	8

$2k$	$\deg(g)$	$\deg(h)$
14	0	8

Table 6.2: Experimental results on $\psi_{2k}g = h$ relationship.

Corollary 13, we get the following result.

Corollary 14 *Consider $g, h \in \mathcal{B}_{2k}$ such that $\psi_{2k}g = h$ where $g \neq 0$. Then either $\deg(h) > k$ or if $h = 0$ then $\deg(g) \geq k$.*

Corollary 14, proves that for $g, h \in \mathcal{B}_{2k}$, there cannot be any relation $\psi_{2k}g = h$, where $\deg(h) = k$. Similar interesting $f * g = h$ relationship has been studied in [20, 4].

From Theorem 12 of chapter 5, the algebraic degree of ψ_n is $2^{\lfloor \log_2 n \rfloor}$ and hence we will always get a constant 1 function g (i.e., of degree 0) such that $\psi_n g = h$, where $\deg(h) = 2^{\lfloor \log_2 n \rfloor}$, i.e., $h = \psi_n$. Similarly extending the result of [20], if $2^t < n \leq 2^{t+1}$, then there always exist $\psi_n g = h$ relations having $\deg(g) = 1$ and $\deg(h) = 2^t + 1$ (the result in [20] shows this only when n is a power of 2). Note that the theoretical results given in [4, Table 4] are not tight due to this reason. In Table 6.2, we present the results in tabular form and this may be compared with Table 6.1. Based on these, it seems that the ψ_{2k} functions have worse profile than ϕ_{2k}. Note that the weight and nonlinearity of ψ_{2k} and ϕ_{2k} are same, but the algebraic degree of ϕ_{2k} is in general greater than that of ψ_{2k}.

A more general class of functions with maximum possible algebraic immunity has been presented in Construction 5 of Chapter 5. For n even we name the function ζ_{2k}, $k \geq 0$ and the exact definition is as following.

Construction 7

$$\zeta_{2k}(x) = \begin{cases} 0 \text{ for } \mathsf{wt}(x) < k, \\ a_x \text{ for } \mathsf{wt}(x) = k, \ a_x \in \{0, 1\}, \\ 1 \text{ for } \mathsf{wt}(x) > k. \end{cases}$$

$2k$	$nl(\zeta_{2k})$	$\deg(\zeta_{2k})$	$\deg(g)$	$\deg(h)$
6	22	5	3	3
			1	4
8	92	7	3	4
			1	5
10	384	9	4	5
			2	6
			2	7
			1	8
$2k$	$nl(\zeta_{2k})$	$\deg(\zeta_{2k})$	$\deg(g)$	$\deg(h)$
12	1584	11	5	6
			3	7
			3	8
			1	9
14	6470	13	6	7
			4	8
			1	9

Table 6.3: Profiles for the functions ζ_{2k}.

Note that if the value of a_x is same for all the weight k inputs x, then it is a symmetric function. However, we will now specifically consider the case where the outputs corresponding to weight k inputs take both the distinct values $0, 1$ and the function becomes non symmetric.

Proposition 20 *Consider ζ_{2k} as described above. Then both $\zeta_{2k}, 1 + \zeta_{2k}$ have minimum degree annihilators at degree k.*

Proof : We already have $\mathsf{AI}_{2k}(\zeta_{2k}) = k$. That both $\zeta_{2k}, 1 + \zeta_{2k}$ has minimum degree annihilators at degree k can be proved considering their weights of $\zeta_{2k}, 1 + \zeta_{2k}$ and following the same kind of argument as in the proof of Theorem 2 in Chapter 3. ■

Based on Proposition 20, it is not clear whether there exists g having $\deg(g) < k$ such that $\zeta_{2k} g = h$, where $\deg(h) = k$. Thus we go for the following experimentation. We use similar kind of functions as described in Chapter 5 as follows.

Construction 8

$$G(x_1,\ldots,x_{2k}) = \begin{cases} 1 \text{ for } \mathsf{wt}(x_1,\ldots,x_{2k}) < k, \\ 0 \text{ for } \mathsf{wt}(x_1,\ldots,x_{2k}) > k, \\ b(x_1,\ldots,x_{2k}) \text{ for } \mathsf{wt}(x_1,\ldots,x_{2k}) = k, \end{cases}$$

where $b(x_1,\ldots,x_{2k})$ is a Maiorana-McFarland type bent function.

1. *If $\mathsf{wt}(G) < 2^{2k-1}$, then we choose $2^{2k-1} - \mathsf{wt}(G)$ points randomly from the inputs having weight k and output 0 of G and toggle those outputs to 1 to get ζ_{2k}.*
2. *If $\mathsf{wt}(G) > 2^{2k-1}$, then we choose $\mathsf{wt}(G) - 2^{2k-1}$ points randomly from the inputs having weight k and output 1 of G and toggle those outputs to 0 to get ζ_{2k}.*

Thus we get balanced ζ_{2k}. As we have already described in Proposition 20, the $f * g = h$ relationships for the functions of the type of ζ_{2k} may not be decided immediately. Thus we present some experimental results in Table 6.3 for this purpose for a randomly chosen ζ_{2k} for each $6 \leq 2k \leq 14$.

6.4 Experimental Results

6.4.1 Rotation Symmetric Functions

We consider the following rotation symmetric functions with good cryptographic properties and full algebraic immunity as they have been studied in Section 3.4.1 of Chapter 3.

First we consider the 7-variable, 2-resilient, nonlinearity 56 rotation symmetric Boolean functions with algebraic immunity 4. There are 12 such functions. For all these functions f, we got $f * g = h$ relationship where g is a linear function and h has degree 4. Thus these functions are not good in resisting fast algebraic attacks.

Next we consider the 8-variable, 1-resilient, nonlinearity 116 rotation symmetric Boolean functions with algebraic immunity 4. There are 6976 such functions. Out of them there are 6080 many functions f, for which we get good profile. For these functions, we get the profile like $\deg(g) = 3, \deg(h) = 4$; $\deg(g) = 2, \deg(h) = 5$ and $\deg(g) = 1, \deg(h) = 6$. In all these cases we fix degree of h and then find the minimum degree g. Thus there exist 8-variable, 1-resilient, nonlinearity 116 rotation symmetric Boolean functions where we get good profile

in terms of fast algebraic attack. Further note that these functions are of degree 6 by itself. The truth table of one of these functions is as below in hexadecimal format:

0005557337726F4A1E6E7B4C3CAB7598
03FD7CB86ADA61F41FE48C9E7A26C280

6.4.2 (Modified) Balanced Patterson-Wiedemann type Functions

Patterson and Wiedemann [139, 140] considered the Boolean functions on odd number of input variables n and succeeded to find out functions having nonlinearity strictly greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd $n \geq 15$. This result is pioneering as this is the first instance when such a high nonlinearity has been demonstrated and further till date there is no other strategy to get such functions. Later in [116] these functions have been changed heuristically to get highly nonlinear balanced functions. We consider one of the functions presented in [116], which is a balanced function on 15 variables having nonlinearity $16262 > 2^{15-1} - 2^{\frac{15-1}{2}}$. We found that the algebraic immunity of the function we have considered is 7 (not 8, which is the maximum possible for 15-variable functions). Given this function f, we experimented on the $f * g = h$ relationships fixing $\deg(h) \geq 7$ and then finding out the minimum degree g. The $(\deg(g), \deg(h))$ relationships for the function f is as follows: (6, 7), (6, 8), (3, 9), (3, 10), (2, 11), (2, 12), (1,13).

6.5 Functions with Additional Constraints over Maximum AI

In this section we consider the functions $f \in \mathcal{B}_n$ with maximum possible AI, i.e., $\lceil \frac{n}{2} \rceil$ with the property that given $fg = h$ relation such that $\deg(h) = \lceil \frac{n}{2} \rceil$, we should get $\deg(g) \geq \lfloor \frac{n}{2} \rfloor$. This is the additional constraint. These functions are indeed better than any functions with only maximum AI with respect to fast algebraic attacks, since one can not get a g having $\deg(g) < \lfloor \frac{n}{2} \rfloor$ when $\deg(h)$ is fixed at $\lceil \frac{n}{2} \rceil$. From Proposition 16, one can see that this is the best possible case when $\deg(h)$ is fixed at $\lceil \frac{n}{2} \rceil$.

First concentrate on functions similar to ϕ_{2k}, ψ_{2k} having full algebraic immunity such that a function $f \in \mathcal{B}_{2k}$, the lowest degree annihilators are at degree k and for its complement $1 + f$, the lowest degree annihilators are at degree $k + 1$. Hence from Corollary 12 that these functions cannot have $fg = h$ relation such that $\deg(h) = k$ and $\deg(g) < k$. Now one can

also check that the $(2k+1)$-variable function $F = x_{2k+1} + f$ is of algebraic immunity $k+1$; further F is balanced. One can also check that the function $x_{2k+2} + x_{2k+1} + f$ has algebraic immunity $k+1$ and it is also an 1-resilient function. We summarize these results below.

Theorem 15

1. *For any even n, it is possible to get unbalanced $f \in \mathcal{B}_n$ with maximum possible* AI *i.e., $\frac{n}{2}$ such that given any $f * g = h$ relation having* $\deg(h) = \frac{n}{2}$, $\deg(g) \not< \frac{n}{2}$.

2. *For any even n it is possible to get* 1*-resilient function having full algebraic immunity.*

With respect to Theorem 15(1), it is open to get such balanced functions f_b when n is even. We solve this problem in Subsection 6.5.1 for all even n except when n is an exact power of 2 and then considering $x_{n+1} + f_b$ the corresponding case for Theorem 15(2) will be solved for $n+1$ (odd) variable functions. Note that experimental evidences of resilient functions with full algebraic immunity are available in Chapter 3, but no theoretical result is available in the literature so far.

The results in Theorem 15 are proved using the functions ϕ_{2k} and ψ_{2k} which are of the property that only one of $f, 1+f$ has minimum degree annihilators at $\mathsf{AI}_{2k}(f)$ and the other one has minimum degree annihilators at degree $1 + \mathsf{AI}_{2k}(f)$. We have already studied for such functions having $\mathsf{wt}(f) = 2^{2k-1} - \binom{2k-1}{k}$ (i.e., these functions are not balanced) and $\mathsf{nl}(f) \leq 2^{2k-1} - \binom{2k-1}{k}$.

6.5.1 Annihilators of $f, 1+f$ at the Same Degree

Now we will concentrate on the functions such that the minimum degree annihilators of the function and its complement are at the same degree but they never cancel out when added. We formally define this as below.

Definition 13 *Suppose $f \in \mathcal{B}_{2k}$ be such that $\mathsf{AI}_{2k}(f) = k$, the maximum possible; the lowest degree annihilators of both f and $1+f$ are at degree k. Further there is no two nonzero k-degree annihilators g and h of f and $1+f$ respectively, such that* $\deg(g+h) < k$. *We denote such functions by P_{2k} functions.*

Theorem 16 *Suppose f be a P_{2k} function. Then*

1. $\mathsf{AI}_{2k+1}(x_{2k+1} + f) = k + 1$, *which is the maximum possible;*
2. *if for* $f_1, f_2 \in \mathcal{B}_{2k}$, $ff_1 = f_2$ *where* $\deg(f_1) \leq \deg(f_2) = k$ *then* $\deg(f_1) = k$;
3. $\mathsf{nl}(f) \geq 2^{2k-1} - \binom{2k-1}{k-1}$.

Proof : Let us denote $F = x_{2k+1} + f$. Any nonzero annihilator of F is of the form $g_1 + x_{2k+1}(g_1 + g_2)$, where $g_1 \in AN(f)$ and $g_2 \in AN(1 + f)$ and both g_1, g_2 are not 0 at the same time. Similarly any nonzero annihilator of $1 + F$ is of the form $g_2 + x_{2k+1}(g_1 + g_2)$. As $g_1 \neq g_2$ and their highest degree terms can not cancel out in $g_1 + g_2$, their degree of the annihilators can not be less than or equal to k. Thus $\mathsf{AI}_{2k+1}(F) = k + 1$.

Now we prove item 2. Consider we have some f_1, f_2 such that $ff_1 = f_2$ with $\deg(f_1) \leq k$, $\deg(f_2) = k$. Note that $ff_1 = f_2$ iff $f(f_1+f_2) = 0$ and $(1+f)f_2 = 0$ [20]. So, $f_1 = (f_1+f_2)+f_2$ is the sum of the two k degree annihilators $f_1 + f_2$ and f_2 of f and $1 + f$ respectively. As their highest degree terms never cancel out we have $\deg(f_1) = k$.

Next we prove the last item. Since $x_{2k+1} + f$ is of full algebraic immunity $k+1$, following Theorem 4 by Lobanov, one gets $\mathsf{nl}(x_{2k+1} + f) \geq 2^{2k} - \binom{2k}{k}$. As for every $2k$-variable function f, we have $\mathsf{nl}(x_{2k+1} + f) = 2\mathsf{nl}(f)$, we get $\mathsf{nl}(f) \geq 2^{2k-1} - \binom{2k-1}{k-1}$. ■

This kind of function provides the best possible relationship when we use functions f on n variables and consider $f * g = h$ relationship with $\deg(h) = \frac{n}{2}$ as in that case $\deg(g)$ can not be less than $\frac{n}{2}$. This is the optimum situation when $\deg(h) = \frac{n}{2}$.

Now consider the function ζ_{2k} from Construction 7. One can get a balanced $\zeta_{2k}(x)$ if the outputs corresponding to half of the weight k inputs are 0 and the outputs corresponding to half of the weight k inputs are 1. Note that there are $\binom{\binom{2k}{k}}{\frac{1}{2}\binom{2k}{k}}$ many balanced functions of the form ζ_{2k}. From $\zeta_{2k}(x)$, the Construction 8 is attempted to get balanced functions.

Now we like to point out the problems with the Constructions 7, 8 where the annihilators of f and $1 + f$ are at the same degree.

1. The constructions are randomized and hence the exact nonlinearity of the functions cannot be calculated. In fact, the experimental results show that the nonlinearity of the functions are slightly less than $2^{2k-1} - \binom{2k-1}{k-1}$.
2. Experimental results in Table 6.3 show that there exists g having $\deg(g) < k$ such that $\zeta_{2k}g = h$, where $\deg(h) = k$.

We solve these problems in the construction presented in the following sub section where the functions will have nonlinearity not less than $2^{2k-1} - \binom{2k-1}{k-1}$ and there cannot be any $\deg(g) < k$.

6.5.2 The Exact Construction

We present the following construction.

Construction 9 *Consider $\eta_{2k} \in \mathcal{B}_{2k}$, as follows:*

$$\eta_{2k}(x) = \begin{cases} 1 \text{ for } \mathsf{wt}(x) < k, \\ a_x \text{ for } \mathsf{wt}(x) = k,\ a_x \in \{0,1\},\ \text{with the constraint } a_x = a_{\overline{x}}, \\ 0 \text{ for } \mathsf{wt}(x) > k, \end{cases}$$

where $\overline{x}$ is the bitwise complement of the vector x. Further all the a_x's are not same, i.e., they take both the values $0, 1$.

Theorem 17 *The functions $\eta_{2k}(x)$ as in Construction 9 are P_{2k} functions.*

Proof : Using the similar proof technique used in Theorem 10 in Chapter 5, one gets that both η_{2k} and $1 + \eta_{2k}$ has no annihilators at degree less than k. Further, $\sum_{i=0}^{k} \binom{n}{i}$ is greater than both $\mathsf{wt}(\eta_{2k})$ and $\mathsf{wt}(1 + \eta_{2k})$ and hence from Theorem 2, both η_{2k} and $1 + \eta_{2k}$ must have annihilator at degree less than or equal to k. Hence both η_{2k} and $1+\eta_{2k}$ have minimum degree annihilators exactly at degree k.

Any k degree function $g \in \mathcal{B}_{2k}$ can be written as

$$a_0 + \sum_{i=0}^{n} a_i x_i + \cdots + \sum_{1 \le < i_1 < \cdots < i_k \le n} a_{i_1,\ldots,i_k} x_{i_1} \ldots x_{i_k},$$

where the coefficients a's are either 0 or 1. If g is an annihilator of η_{2k} then $g(x) = 0$ when $\eta_{2k}(x) = 1$. Since $\eta_{2k}(x) = 1$ for $\mathsf{wt}(x) < k$, we can eliminate all the coefficients (a's) associated to monomials of degree less than or equal to $k-1$ of g. Then we have $\eta_{2k}(x) = 1$ for some input vectors x of weight k. For such an $x = (b_1, \ldots, b_n)$, where $b_{i_1} = \cdots = b_{i_k} = 1$ and rest are 0, one can eliminate the coefficient $a_{i_1,\ldots,i_k}$. Thus the k degree independent annihilators of η_{2k} form the set $S_1 = \{x_{j_1} \ldots x_{j_k} \mid \eta_{2k}(b_1, \ldots, b_n) = 0 \text{ and } b_{j_1} = \cdots = b_{j_k} = 1, \text{ rest are } 0\}$. Here any k-degree annihilator of η_{2k} does not contain any monomial of degree less than k.

Define $f'(x) = 1 + \eta_{2k}(\overline{x})$. Following the similar proof for $\eta_{2k}(x)$, one can prove that the space of k degree annihilators of f' is generated by the basis set $\{x_{j_1} \dots x_{j_k} \mid f'(b_1, \dots, b_n) = 0$ and $b_{j_1} = \dots = b_{j_k} = 1$, rest are $0\}$. Hence, the k degree annihilator space of $f'(\overline{x}) = 1 + \eta_{2k}(x)$ is generated by the basis set $\{(1 + x_{j_1}) \dots (1 + x_{j_k}) \mid f'(1 + b_1, \dots, 1 + b_n) = 1 + \eta_{2k}(b_1, \dots, b_n) = 0$ and $b_{j_1} = \dots = b_{j_k} = 0$, rest are $1\}$. So, the subspace of k degree monomials of k degree annihilators of $1 + \eta_{2k}$ is generated by the basis set $S_2 = \{x_{j_1} \dots x_{j_k} \mid \eta_{2k}(b_1, \dots, b_n) = 1$ and $b_{j_1} = \dots = b_{j_k} = 0$, rest are $1\}$. One can check that these two sets S_1 and S_2 are disjoint iff $\eta_{2k}(x) = \eta_{2k}(\overline{x})$ for $wt(x) = k$.

Since the basis sets S_1, S_2 are disjoint, the k degree terms of any annihilator of η_{2k} and the k degree terms of any annihilator of $1 + \eta_{2k}$ cannot be the same. Thus the proof. ■

Corollary 15 *One can get a balanced η_{2k} iff $2k$ is not a power of 2 and the count of such balanced functions is $\binom{\frac{1}{2}\binom{2k}{k}}{\frac{1}{4}\binom{2k}{k}}$.*

Proof : For a $2k$-variable function, there are $\binom{2k}{k}$ many input vectors of weight k and there are $\frac{1}{2}\binom{2k}{k}$ many $(x, \overline{x})$ distinct pairs of weight k. One can construct a balanced η_{2k} if and only if $\frac{1}{2}\binom{2k}{k}$ is even, i.e., $\binom{2k}{k}$ is divisible by 4. Since $\binom{2k}{k} = 2\binom{2k-1}{k-1}$, we need to test whether $\binom{2k-1}{k-1}$ is even.

Suppose the $t = \lfloor \log_2 2k \rfloor + 1$ bit binary representations of $2k, k, 2k-1$ and $k-1$ are as follows (most significant bit at the left most position):

$$\begin{array}{lcccccccccc}
2k & = & b_t & b_{t-1} & \dots & b_{l+1} & b_l = 1 & 0 & 0 & \dots & 0, \\
k & = & 0 & b_t & \dots & b_{l+2} & b_{l+1} & b_l = 1 & 0 & \dots & 0, \\
2k-1 & = & b_t & b_{t-1} & \dots & b_{l+1} & 1 + b_l = 0 & 1 & 1 & \dots & 1, \\
k-1 & = & 0 & b_t & \dots & b_{l+2} & b_{l+1} & 1 + b_l = 0 & 1 & \dots & 1,
\end{array}$$

where $1 < l \leq t$, $b_i \in \{0, 1\}$ and $b_t = b_l = 1$. Now following Lucas' theorem [45, Page 79] with the prime 2, we have $\binom{2k-1}{k-1} \equiv \binom{b_t}{0}\binom{b_{t-1}}{b_t} \dots \binom{0}{b_{l+1}}\binom{1}{0}\binom{1}{1} \dots \binom{1}{1} \bmod 2$. If $2k$ is a power of 2, then $t = l$. So, $\binom{2k-1}{k-1} \equiv \binom{1}{0}\binom{1}{1} \dots \binom{1}{1} \bmod 2$, i.e., $\binom{2k-1}{k-1} \equiv 1 \bmod 2$. Hence $\binom{2k-1}{k-1}$ is odd.

If $2k$ is not a power of 2, then $\binom{2k-1}{k-1} \equiv \binom{b_{t-1}}{b_t} \dots \binom{b_{l+1}}{b_{l+2}}\binom{0}{b_{l+1}} \bmod 2$. At some place we will get $b_s = 0$ and $b_{s+1} = 1$ for $l \leq s < t$ because $b_t = 1$. Hence $\binom{2k-1}{k-1}$ is even if $2k$ is not a power of 2.

Thus $\binom{2k}{k}$ is divisible by 4, when $2k$ is not exactly a power of 2. In such a case, there will be $\frac{1}{2}\binom{2k}{k}$ many distinct pairs of $(x, \overline{x})$, where x is a $2k$ bit binary pattern of weight k. One can choose $\frac{1}{4}\binom{2k}{k}$ many distinct pairs and in such inputs of η_{2k}, output 1 is assigned

and for the rest of $\frac{1}{4}\binom{2k}{k}$ many distinct pairs of inputs, output 0 is assigned. This provides a balanced η_{2k}. Note that the number of such distinct balanced η_{2k} is $\binom{\frac{1}{2}\binom{2k}{k}}{\frac{1}{4}\binom{2k}{k}}$. ■

Now an important question is whether there exist balanced P_{2k} functions when $2k$ is a power of 2. We have checked that for $2k = 4 = 2^2$, there is no balanced P_4 function by running exhaustive computer program.

6.5.3 Functions on Odd Number of Input Variables

Now let us study the functions f on odd number of input variables $2k+1$ having maximum possible algebraic immunity $k+1$. That is the functions must be balanced. Consider the following balanced symmetric functions from Chapter 5 on $2k+1$ variables having full algebraic immunity $k+1$.

Construction 10 *Consider* $\tau_{2k+1} \in \mathcal{B}_{2k+1}$, *as follows:*

$$\tau_{2k+1}(x) = \begin{cases} 1 \text{ for } \mathsf{wt}(x) \leq k, \\ 0 \text{ for } \mathsf{wt}(x) \geq k+1. \end{cases}$$

We list a few experimental values of minimum degree of g when $\tau_{2k+1}g = h$ and $\deg(h) = k+1$. In the format $< 2k+1, \deg(g), \deg(h) >$ these values are $< 5, 1, 3 >$, $< 7, 1, 4 >$, $< 9, 1, 5 >$, $< 11, 2, 6 >$. Note that the minimum degree of g is substantially less than k and hence the functions τ_{2k+1} are not interesting in resistance against fast algebraic attacks.

To get a better resistance against fast algebraic attack, we are interested about the balanced functions with the following additional property. Given any $f * g = h$ relation having $\deg(h) = k+1$, we require that $\deg(g) \geq k$.

We run exhaustive search for $2k+1 = 5$ variable functions and found such functions. One example is the truth table 00000001000101110001101111011111 which is of nonlinearity 10 and algebraic degree 4. Note that there is no nonlinearity 12 function on 5 variables with such property. Existence of such functions for 7 variables onwards is an open question.

6.6 Conclusion

In this chapter, construction of balanced Boolean functions with maximum possible algebraic immunity is studied with an additional property which is necessary to resist certain kind

of fast algebraic attacks. We have studied (in some cases theoretically, in some other cases experimentally) a few existing constructions of Boolean functions for their resistances against certain kinds of fast algebraic attacks. The additional property considered here is, given an n-variable (n even) balanced function f with maximum possible AI, i.e., $\frac{n}{2}$, and given two n-variable Boolean functions g, h such that $f * g = h$, if $\deg(h) = \frac{n}{2}$, then $\deg(g)$ must be greater than or equal to $\frac{n}{2}$. Our results can also be used to present theoretical construction of resilient Boolean functions having maximum possible AI. Getting a primary construction of cryptographically significant Boolean functions (mainly with high nonlinearity) having maximum possible algebraic immunity and good resistance against fast algebraic attacks has not been solved satisfactorily yet.

Chapter 7

Reducing the Number of Homogeneous Linear Equations in Finding Annihilators

Results on algebraic attacks have received a lot of attention recently in studying the security of crypto systems [3, 7, 21, 26, 41, 43, 58, 56, 51, 109, 4, 69, 54]. Boolean functions are important primitives to be used in the crypto systems and in view of the algebraic attacks, the annihilators of a Boolean function play considerably serious role [33, 22, 123, 134].

It is known from [56, 123] that for any function f or $1 + f$ must have an annihilator at the degree $\lceil \frac{n}{2} \rceil$. Thus the target of a good design is to use a function f such that neither f nor $1 + f$ has an annihilator at a degree less than $\lceil \frac{n}{2} \rceil$. Thus there is a need to construct such functions and the first one in this direction is described in Chapter 4. Later symmetric functions with this property has been presented in Chapter 6 (also independently in [22]). However, all these constructions are not rich in terms of some important cryptographic properties like nonlinearity, resiliency etc.

Thus there is a need to study the Boolean functions, which are rich in terms of other cryptographic properties, in terms of their annihilators. One has to find out the annihilators of a given Boolean function for this. Moreover, for Cryptanalysis, it is necessary to find the annihilators. Initially a basic algorithm in finding the annihilators has been proposed in [123, Algorithms 1 and 2]. A minor modification of [123, Algorithm 2] has been presented very recently in [20] to find out relationships for algebraic and fast algebraic attacks. In [22], there is an efficient algorithm to find the annihilators of symmetric Boolean functions, but

symmetric Boolean functions are not cryptographically promising. Algorithms using Gröbner bases are also interesting in this area [6], but still they are not considerably efficient. Recently more efficient algorithms have been designed in this direction [4, 69]. The algorithm presented in [4] can be used efficiently to find out relationships for algebraic and fast algebraic attacks. In [4], matrix triangularization has been exploited nicely to solve the annihilator finding problem (of degree d for an n-variable function) in $O(\binom{n}{d}^2)$ time complexity. In [69] a probabilistic algorithm having time complexity $O(n^d)$ has been proposed where the function is divided to its sub functions recursively and the annihilators of the sub functions are checked to study the annihilators of the original function.

The main idea in this chapter is to reduce the size of the matrix (used to solve the system of homogeneous linear equations) as far as possible. We could successfully improve the handling of equations associated with small weight inputs of the Boolean function. This uses certain structure of the matrix that we discover here. We start with a matrix $M_{n,d}(g)$ (see Theorem 18) which is self inverse and its discovered structure allows to compute the new equations efficiently by considering the matrix UA^r (see Theorem 20 in Section 7.2). Moreover, each equation associated with a low weight input point directly provides the value of an unknown coefficient of the annihilator, which is the key point that allows to lower the number of unknowns. Further reduction in the size of the matrix is dependent on getting a proper linear transformation on the input variables of the Boolean function, which is discussed in Section 7.3.

One may wonder whether the very recently available strategies in [4, 69] can be applied after the initial reduction proposed in this paper to get further improvements in finding the lowest degree annihilators. The standard Gaussian reduction technique ([69, Algorithm 1]) is used in the main algorithm [69, Algorithm 2], and in that case our idea of reduction of the matrix size will surely provide improvement. However, the ideas presented in [4, Algorithm 1, 2] and [69, Algorithm 3] already exploit the structure of the linear system in an efficient way. In particular, the algorithms in [4] by themselves deal with the equations of small weight efficiently. Thus it is not clear whether the reduction of matrix size proposed by us can be applied to exploit further efficiency from these algorithms.

7.1 Preliminaries

Consider all the n-variable Boolean functions of degree at most d, i.e., $\mathcal{R}(n, d)$, the Reed-Muller code of order d and length 2^n. Any Boolean function can be seen as a multivariate

polynomial over $\mathbb{F}_2$. Note that $\mathcal{R}(n,d)$ is a vector subspace of the vector space $\mathcal{B}_n$ (which is $\mathcal{R}(n,n)$), the set of all n-variable Boolean functions. As the elements of $\mathcal{R}(n,d)$ are multivariate polynomials over $\mathbb{F}_2$, then the standard basis is the set of all nonzero monomials of degree less than or equal to d. That is, the standard basis is

$$S_{n,d} = \{x_{i_1} \dots x_{i_l} \mid 1 \leq l \leq d \text{ and } 1 \leq i_1 < i_2 < \cdots < i_l \leq n\} \cup \{1\},$$

where the input variables of the Boolean functions are $x_1, \dots, x_n$.

The ordering among the monomials is considered in lexicographic ordering ($<^l$) as usual, i.e., $x_{i_1}x_{i_2}\dots x_{i_k} <^l x_{j_1}x_{j_2}\dots x_{j_l}$ if either $k < l$ or $k = l$ and there is $1 \leq p \leq k$ such that $i_k = j_k$, $i_{k-1} = j_{k-1}, \dots, i_{p+1} = j_{p+1}$ and $i_p < j_p$. For example, for $n = 7$, $x_1x_3x_6 <^l x_1x_2x_4x_5$ and $x_1x_3x_6 <^l x_1x_4x_6$. So, the set $S_{n,d}$ is a totally ordered set with respect to this lexicographical ordering ($<^l$). Using this ordering we refer the monomials according their order, i.e., the k-th monomial as m_k, $1 \leq k \leq \sum_{i=0}^{d} \binom{n}{i}$ following the convention $m_l <^l m_k$ if $l < k$.

Definition 14 *Given $n > 0$, $0 \leq d \leq n$, we define a mapping*

$$v_{n,d} : \mathbb{F}_2^n \mapsto \mathbb{F}_2^{\sum_{i=0}^{d} \binom{n}{i}},$$

such that $v_{n,d}(x) = (m_1(x), m_2(x), \dots, m_{\sum_{i=0}^{d} \binom{n}{i}}(x))$. Here $m_i(x)$ is the ith monomial as in the lexicographical ordering ($<^l$) evaluated at the point $x = (x_1, x_2, \dots, x_n)$.

To evaluate the value of the t-th coordinate of $v_{n,d}(x_1, x_2, \dots, x_n)$ for $1 \leq t \leq \sum_{i=0}^{d} \binom{n}{i}$, i.e., $[v_{n,d}(x_1, \dots, x_n)]_t$, one requires to calculate the value of the monomial m_t (either 0 or 1) at $(x_1, x_2, \dots, x_n)$. Now we define a matrix $M_{n,d}$ with respect to a n-variable function f. To define this we use the ordering $<_l$ over the elements of vector space $\mathbb{F}_2^n$ (see Chapter 2).

Definition 15 *Given $n > 0$, $0 \leq d \leq n$ and an n-variable Boolean function f, we define a $\mathsf{wt}(f) \times \sum_{i=0}^{d} \binom{n}{i}$ matrix*

$$M_{n,d}(f) = \begin{bmatrix} v_{n,d}(X_1) \\ v_{n,d}(X_2) \\ \vdots \\ v_{n,d}(X_{\mathsf{wt}(f)}) \end{bmatrix}$$

where $X_i \in supp(f)$, $1 \leq i \leq \mathsf{wt}(f)$ and $X_1 <_l X_2 <_l \cdots <_l X_{\mathsf{wt}(f)}$.

Note that the matrix $M_{n,d}(f)$ is the transpose of the restricted generator matrix for Reed-Muller code of length 2^n and order d, $\mathcal{R}(d,n)$, to the support of f (see also [26, Page 7]). Any row of the matrix $M_{n,d}(f)$ corresponding to an input vector $(x_1,\ldots,x_n)$ is

$$\overbrace{1}^{0\,\deg}\ \overbrace{x_1,\ \ldots,\ x_i,\ \ldots,\ x_n}^{1\,\deg}\ \ldots\ \overbrace{x_1\ldots x_d,\ \ldots,\ x_{i_1}\ldots x_{i_d},\ \ldots,\ x_{n-d+1}\ldots x_n}^{d\,\deg}.$$

Each column of the matrix is represented by a specific monomial and each entry of the column tells whether that monomial is satisfied by the input vector which identifies the row, i.e., the rows of this matrix correspond to the evaluations of the monomials having degree at most d on support of f. As already discussed, here we have one-to-one correspondence from the input vectors $x=(x_1,\ldots,x_n)$ to the row vectors $v_{n,d}(x)$ of length $\sum_{i=0}^{d}\binom{n}{i}$. So, each row is fixed by an input vector.

7.1.1 Annihilators of f and Rank of the Matrix $M_{n,d}(f)$

Let f be an n-variable Boolean function. We are interested to find out the lowest degree annihilators of f. Let a nonzero $g\in\mathcal{B}_n$ be an annihilator of f, i.e., $f(x_1,\ldots,x_n)*g(x_1,\ldots,x_n)=0$ for all $(x_1,\ldots,x_n)\in\mathbb{F}_2^n$. In terms of truth table, this means that the function $f\ AND\ g$ will be a constant zero function, i.e., for each vector $(x_1,\ldots,x_n)\in\mathbb{F}_2^n$, the output of $f\ AND\ g$ will be zero. That means,

$$g(x_1,\ldots,x_n)=0 \text{ if } f(x_1,\ldots,x_n)=1. \tag{7.1}$$

Suppose degree of the function g is less than equal to d, then the ANF of g is of the form

$$g(x_1,\ldots,x_n)=a_0+\sum_{i=0}^{n}a_ix_i+\cdots+\sum_{1\le i_1<i_2\cdots<i_d\le n}a_{i_1,\ldots,i_d}x_{i_1}\cdots x_{i_d}$$

where the subscripted a's (coefficients of monomials) are from $\mathbb{F}_2$ and not all of them are zero. Following Equation 7.1, we get the following $\mathsf{wt}(f)$ many homogeneous linear equations

$$a_0+\sum_{i=0}^{n}a_ix_i+\cdots+\sum_{1\le i_1<i_2\cdots<i_d\le n}a_{i_1,\ldots,i_d}x_{i_1}\cdots x_{i_d}=0, \tag{7.2}$$

considering the input vectors $(x_1,\ldots,x_n)\in supp(f)$. This is a system of homogeneous linear equations on a's with $\sum_{i=0}^{d}\binom{n}{i}$ many a's as variables. The matrix form of this system of equations is $M_{n,d}(f)\ A^{tr}=O$, where $A=(a_0,a_1,a_2,\ldots,a_{n-d+1,\ldots,n})$, the row vector of

coefficients of the monomials which are ordered according to the lexicographical order $<^l$. Each nonzero solution of the system of equations formed by Equation 7.2 gives an annihilator g of degree less than or equal to d. This is basically the Algorithm 1 presented in [123]. Since the number of solutions of this system of equations are connected to the rank of the matrix $M_{n,d}(f)$, it is worth to study the rank and the set of linear independent rows/columns of matrix $M_{n,d}(f)$. If the rank of matrix $M_{n,d}(f)$ is equal to $\sum_{i=0}^{d}\binom{n}{i}$ (i.e., number of columns) then the only solution is the zero solution. So, for this case f has no annihilator of degree less than or equal to d. This implies that the number of rows is greater than or equal to the number of columns, i.e., $\mathsf{wt}(f) \geq \sum_{i=0}^{d}\binom{n}{i}$ which is the Theorem 2 in Chapter 3. If the rank of the matrix $M_{n,d}(f)$ is equal to $\sum_{i=0}^{d}\binom{n}{i} - k$ for $k > 0$ then the number of linearly independent solutions of the system of equations is k which gives k many linearly independent annihilators of degree less than or equal to d and $2^k - 1$ many number of annihilators of degree less than or equal to d. However, to implement algebraic attack one needs only linearly independent annihilators. Hence, finding the degree of lowest degree annihilator of f and $1 + f$, one can use the following algorithm.

Algorithm 3
for$(i = 1$ *to* $\lceil \frac{n}{2} \rceil - 1)$ {
    find the rank r_1 *of the matrix* $M_{n,i}(f)$*;*
    find the rank r_2 *of the matrix* $M_{n,i}(1 + f)$*;*
    if $\min\{r_1, r_2\} < \sum_{j=0}^{i}\binom{n}{j}$ *then output* i*;*
}
output $\lceil \frac{n}{2} \rceil$*;*

Since either f or $1+f$ has an annihilator of degree less than or equal to $\lceil \frac{n}{2} \rceil$, we are interested only to check till $i = \lceil \frac{n}{2} \rceil$. This algorithm is equivalent to Algorithm 1 in [123].

The simplest and immediate way to solve the system of these equations or find out the rank of $M_{n,d}(f), M_{n,d}(1 + f)$ is the Gaussian elimination process. To check the existence or to enumerate the annihilators of degree less than or equal to $\lceil \frac{n}{2} \rceil$ for a balanced function, the complexity is approximately $(2^{n-2})^3$. Considering this time complexity, it is not encouraging to check annihilators of a function of 20 variables or more using the presently available computing power. However, given n and d, the matrix $M_{n,d}(f)$ has pretty good structure, which we explore in this chapter towards a better algorithm (that is solving the set of homogeneous linear equations in an efficient way by decreasing the size of the matrix involved).

7.2 Faster Strategy to Construct the Set of Homogeneous Linear Equations

In this section we present an efficient strategy to reduce the set of homogeneous linear equations. First we present a technical result.

Theorem 18 *Let g be an n-variable Boolean function defined as $g(x) = 1$ iff $\mathsf{wt}(x) \leq d$ for $0 \leq d \leq n$. Then $M_{n,d}(g)^{-1} = M_{n,d}(g)$, i.e., $M_{n,d}(g)$ is a self inverse matrix.*

Proof : Suppose $\mathcal{F} = M_{n,d}(g)M_{n,d}(g)$. Then the i-th row and j-th column entry of $\mathcal{F}$ (denoted by $\mathcal{F}_{i,j}$) is the scalar product of i-th row and j-th column of $M_{n,d}(g)$. Suppose the i-th row is $v_{n,d}(x)$ for $x \in \{0,1\}^n$ having $x_{q_1}, \ldots, x_{q_l}$ as 1 and others are 0. Further consider that the j-th column is the evaluation of the monomial $x_{r_1} \ldots x_{r_k}$ at the input vectors belonging to the support of g. If $\{r_1, \ldots, r_k\} \not\subseteq \{q_1, \ldots, q_l\}$ then $\mathcal{F}_{ij} = 0$. Otherwise, $\mathcal{F}_{i,j} = \binom{l-k}{0} + \binom{l-k}{1} + \cdots + \binom{l-k}{l-k} \bmod 2 = 2^{l-k} \bmod 2$. So, $\mathcal{F}_{i,j} = 1$ iff $l = k$, i.e, $\{x_{r_1}, \ldots, x_{r_k}\} = \{x_{q_1}, \ldots, x_{q_l}\}$. That implies, $\mathcal{F}_{i,j} = 1$ iff $i = j$, i.e., $\mathcal{F}$ is identity matrix. Hence, $M_{n,d}(g)$ is its own inverse. ■

See the following example for the structure of $M_{n,d}(g)$ when $n = 4$ and $d = 2$.

Example 7 *Let us present an example of $M_{n,d}(g)$ for $n = 4$ and $d = 2$. We have $\{1, x_1, x_2, x_3, x_4, x_1x_2, x_1x_3, x_2x_3, x_1x_4, x_2x_4, x_3x_4\}$, the list of 4-variable monomials of degree less than or equal to 2 in ascending order ($<^l$).*

Similarly, $\{(0,0,0,0), (1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1), (1,1,0,0), (1,0,1,0), (0,1,1,0), (1,0,0,1), (0,1,0,1), (0,0,1,1)\}$ present the 4 dimensional vectors of weight less than or equal to 2 in ascending order ($<_l$). So the matrix

$$M_{4,2}(g) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

One may check that $M_{4,2}(g)$ is its inverse.

Lemma 9 *Let A be a nonsingular $m \times m$ binary matrix where the row vectors are denoted as $v_1, v_2, \ldots, v_m$. Let U be a $k \times m$ binary matrix, $k \leq m$, where the vectors are denoted as $u_1, u_2, \ldots, u_k$. Let $W = UA^{-1}$, a $k \times m$ binary matrix. Consider that a matrix A' is formed from A by replacing the rows $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ of A by the vectors $u_1, u_2, \ldots, u_k$. Further consider that a $k \times k$ matrix W' is formed by taking the i_1-th, i_2-th, $\ldots$, i_k-th columns of W (out of m columns). Then A' is nonsingular iff W' is nonsingular.*

Proof : Without loss of generality, we can take $i_1 = 1, i_2 = 2, \ldots, i_k = k$. So, the row vectors of A' are $u_1, \ldots, u_k, v_{k+1}, \ldots, v_m$.

We first prove that if the row vectors of A' are not linearly independent then the row vectors of W' are also not linearly independent. As the row vectors of A' are not linearly independent, we have $\alpha_1, \alpha_2, \ldots, \alpha_m \in \{0, 1\}$ (not all zero) such that $\sum_{i=1}^{k} \alpha_i u_i + \sum_{i=k+1}^{m} \alpha_i v_i = 0$. If $\alpha_i = 0$ for all i, $1 \leq i \leq k$ then $\sum_{i=k+1}^{m} \alpha_i v_i = 0$ which implies $\alpha_i = 0$ for all i, $k+1 \leq i \leq m$ as $v_{k+1}, v_{k+2}, \ldots, v_m$ are linearly independent. So, all α_i's for $1 \leq i \leq k$ can not be zero.

Further, we have $UA^{-1} = W$, i.e., $U = WA$, i.e.,

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix}, \text{ i.e., } u_i = w_i \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix}.$$

$$\text{Hence, } \sum_{i=1}^{k} \alpha_i u_i = \sum_{i=1}^{k} \alpha_i w_i \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix} = r \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix}$$

where $r = (r_1, r_2, \ldots, r_m) = \sum_{i=1}^{k} \alpha_i w_i$.

If the restricted matrix W' were nonsingular, the vector $r' = (r_1, r_2, \ldots, r_k)$ is non zero as $(\alpha_1, \alpha_2, \ldots, \alpha_k)$ is not all zero. Hence, $\sum_{i=1}^{k} \alpha_i u_i + \sum_{i=k+1}^{m} \alpha_i v_i = 0$, *i.e.*, $\sum_{i=1}^{m} r_i v_i + \sum_{i=k+1}^{m} \alpha_i v_i = 0$, *i.e.*, $\sum_{i=1}^{k} r_i v_i + \sum_{i=k+1}^{m} (r_i + \alpha_i) v_i = 0$. This contradicts that $v_1, v_2, \ldots, v_m$ are linearly independent as $r' = (r_1, r_2, \ldots, r_k)$ is nonzero. Hence W' must be singular. This proves one direction.

On the other direction if the restricted matrix W' is singular then there are $\beta_1, \beta_2, \ldots, \beta_k$ not all zero such that $\sum_{i=0}^{k} \beta_i w_i = (0, \ldots, 0, s_{k+1}, \ldots, s_m)$. Hence,

$$\sum_{i=0}^{k} \beta_i u_i = \sum_{i=1}^{k} \beta_i w_i \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix} = s_{k+1}v_{k+1} + \cdots + s_m v_m,$$

i.e., $\sum_{i=0}^{k} \beta_i u_i + \sum_{i=k+1}^{m} s_i v_i = 0$ which says matrix A' is singular. ∎

Following Lemma 9, one can verify the nonsingularity of the larger matrix A' by verifing the nonsingularity of the reduced matrix W'. Thus checking the nonsingularity of the larger matrix A' will be more efficient if the computation of inverse of matrix A, i.e., A^{-1} and matrix product $W = UA^{-1}$ can be done efficiently. The self inverse nature of the matrix $M_{n,d}(g)$ presented in Theorem 18 helps to achieve this efficiency. In the rest of this section we will study this in detail. In the following result we present the Lemma 9 in more general form.

Theorem 19 *Let A be a nonsingular $m \times m$ binary matrix with m-dimensional row vectors $v_1, v_2, \ldots, v_m$ and U be a $k \times m$ binary matrix with m-dimensional row vectors $u_1, u_2, \ldots, u_k$. Consider $W = UA^{-1}$, a $k \times m$ matrix. The matrix A', formed from A by removing the rows $v_{i_1}, v_{i_2}, \ldots, v_{i_l}$ $(l \leq m)$ from A and adding the rows $u_1, u_2, \ldots, u_k$ $(k \geq l)$, is of rank m iff the rank of restricted $k \times l$ matrix W' including only the i_1-th, i_2-th, $\ldots$, i_l-th columns of W is l.*

Proof : Here, the rank of matrix W' is l. So, there are l many rows of W', say $w'_{p_1}, \ldots, w'_{p_l}$ which are linearly independent. So, following the Lemma 9 we have the matrix A'' formed by replacing the rows $v_{i_1}, \ldots, v_{i_l}$ of A by $u_{p_1}, \ldots, u_{p_l}$ is nonsingular, i.e., rank is m. Hence the matrix A' where some more rows are added to A'' has rank m. The other direction can also be shown similar to the proof of the other direction in Lemma 9. ∎

Now using Theorem 18 and Theorem 19, we describe a faster strategy to check the existence of annihilators of certain degree d of a Boolean function f. Suppose g be the Boolean function described in Theorem 18, i.e., $supp(g) = \{x | 0 \leq \mathsf{wt}(x) \leq d\}$. In Theorem 18, we have already shown that $M_{n,d}(g)$ is nonsingular matrix (in fact it is self inverse). Let $\{x \mid \mathsf{wt}(x) \leq d \text{ and } f(x) = 0\} = \{x_1, x_2, \ldots, x_l\}$ and $\{x \mid \mathsf{wt}(x) > d \text{ and } f(x) = 1\} = \{y_1, y_2, \ldots, y_k\}$. If $k < l$, directly one can say that f has annihilator of degree less than or equal to d. Thus, we consider $k \geq l$. Then we consider $M_{n,d}(f)$ as A, $v_{n,d}(x_1), \ldots, v_{n,d}(x_l)$

as $v_{i_1}, \ldots, v_{i_l}$ and $v_{n,d}(y_1)$, $\ldots$, $v_{n,d}(y_k)$ as $u_1, \ldots, u_k$. Then following Theorem 19 we can ensure whether $M_{n,d}(f)$ is nonsingular. If it is nonsingular, then there is no annihilator of degree less than or equal to d, else there are annihilator(s). We may write this in a more concrete form as the following corollary to Theorem 19.

Corollary 16 *Let f be an n-variable Boolean function. Let A^r be the restricted matrix of $A = M_{n,d}(g)$, by taking the columns corresponding to the monomials $x_{i_1}x_{i_2}\ldots x_{i_l}$ such that $l \leq d$ and $f(x) = 0$ when $x_{i_1} = 1, x_{i_2} = 1, \ldots, x_{i_l} = 1$ and rest of the input variables are 0. Further $U = \begin{pmatrix} v_{n,d}(y_1) \\ v_{n,d}(y_2) \\ \vdots \\ v_{n,d}(y_k) \end{pmatrix}$, where $\{y_1, \ldots, y_k\} = \{x \mid \mathsf{wt}(x) > d \text{ and } f(x) = 1\}$. If rank of UA^r is l then there is no annihilator of degree less than or equal to d, else there are annihilator(s) of degree less than or equal to d.*

Proof : As per Theorem 19, here $W = UA^{-1} = UA$, since A is its own inverse following Theorem 18 and hence W' is basically UA^r. Thus the proof follows. ∎

Now we can use the following technique for fast computation of the matrix multiplication UA^r. For this we first present a technical result and its proof is similar in the line of the proof of Theorem 18.

Proposition 21 *Consider g as in Theorem 18. Let $y \in \mathbb{F}_2^n$ such that i_1, i_2, $\ldots$, i_p-th places are 1 and other places are 0. Consider the j-th monomial $m_j = x_{j_1}x_{j_2}\ldots x_{j_q}$ according the ordering $<^l$. Then the j-th entry of $v_{n,d}(y)M_{n,d}(g)$ is 0 if $\{j_1, \ldots, j_q\} \not\subseteq \{i_1, \ldots, i_p\}$ else the value is $\sum_{i=0}^{d-q} \binom{p-q}{i} \bmod 2$.*

Following Proposition 21, we can get each row of U as some $v_{n,d}(y)$ and each column of A^r as m_j and construct the matrix UA^r. One can precompute the sums $\sum_{i=0}^{d-q} \binom{p-q}{i} \bmod 2$ for $d+1 \leq p \leq n$ and $0 \leq q \leq d$, and store them and the total complexity for calculating them is $O(d^2(n-d))$. These sums will be used to fill up the matrix UA^r which is an $l \times k$ matrix according to Corollary 16. Let us denote $\mu_f = |\{x \mid \mathsf{wt}(x) \leq d, f(x) = 1\}|$ and $\nu_f = |\{x \mid \mathsf{wt}(x) > d, f(x) = 1\}|$. Then $\mathsf{wt}(f) = \mu_f + \nu_f$ and the matrix UA^r is of dimension $\nu_f \times (\sum_{i=0}^{d} \binom{n}{i} - \mu_f)$. Clearly $O(d^2(n-d))$ can be neglected with respect to $\nu_f \times (\sum_{i=0}^{d} \binom{n}{i} - \mu_f)$. Thus we have the following result.

Theorem 20 *Consider U and A^r as in Corollary 16. The time (and also space) complexity to construct the matrix UA^r is of the order of $\nu_f \times (\sum_{i=0}^{d} \binom{n}{i} - \mu_f)$. Further checking the*

rank of UA^r (as given in Corollary 16) one can decide whether f has an annihilator at degree d or not.

In fact, to check the rank of the matrix UA^r using Gaussian elimination process, we need not store the ν_f many rows at the same time. One can add one row (following the calculation to compute a row of the matrix given in Proposition 21) at a time incrementally to the previously stored linearly independent rows by checking whether the present row is linearly independent with respect to the already stored rows. If the current row is linearly independent with the existing ones, then we do row operations and add the new row to the previously stored matrix. Otherwise we reject the new row. Hence, our matrix size never crosses the size $(\sum_{i=0}^{d}\binom{n}{i}-\mu_f)\times(\sum_{i=0}^{d}\binom{n}{i}-\mu_f)$.

If ν_f (the number of rows) is less than $(\sum_{i=0}^{d}\binom{n}{i}-\mu_f)$ (the number of variables), then there will be nontrivial solutions and we can directly say that the annihilators exist. Thus we always need to concentrate on the case $\nu_f \geq (\sum_{i=0}^{d}\binom{n}{i}-\mu_f)$, where the matrix size $(\sum_{i=0}^{d}\binom{n}{i}-\mu_f)\times(\sum_{i=0}^{d}\binom{n}{i}-\mu_f)$ provides a further reduction than the matrix size $\nu_f \times (\sum_{i=0}^{d}\binom{n}{i}-\mu_f)$ and one can save more space. This will be very helpful when one tries to check the annihilators of small degree d.

Refer to Subsection 7.2.1 below for detailed description that this algorithm provides asymptotic improvement than [123] in terms of constructing this reduced set of homogeneous linear equations. In terms of the overall algorithm to find the annihilators, our algorithm works around eight times further than [123] in general. Using our strategy to find the reduced matrix first and then using the standard Gaussian elimination technique, we could find the annihilators of any random balanced Boolean functions on 16 variables in around 2 hours in a Pentium 4 personal computer with 1 GB RAM. Note that, the very recently known efficient algorithms [4, 69] can work till 20 variables.

7.2.1 Comparison with Meier et. al. Algorithm

Here we compare the time and space complexity of our strategy with [123, Algorithm 2]. In paper [123], Algorithm 2 is probabilistic. In this section we study the time and space complexity of the algorithm along with it's deterministic version. Using these algorithms we check whether there exist annihilators of degree less than or equal to d of an n-variable function f. As we have already described, ANF of any n-variable function g of degree d is

of the form

$$g(x_1, \ldots, x_n) = a_0 + \sum_{i=0}^{n} a_i x_i + \cdots + \sum_{1 \le i_1 < i_2 \cdots < i_d \le n} a_{i_1,\ldots,i_d} x_{i_1} \cdots x_{i_d}$$

where subscripted a's are from $\mathbb{F}_2$. First we present the exact probabilistic algorithm [123, Algorithm 2].

Algorithm 4

1. *Initialize weight $w = 0$.*
2. *For all x's of weight w with $f(x) = 1$, substitute each x in $g(x) = 0$ to derive a linear equation on the coefficients of g, with a single coefficient of weight w. Use this equation to express this coefficient iteratively by coefficients of lower weight.*
3. *If $w < d$, increment w by 1 and go to step 2.*
4. *Choose random arguments x of arbitrary weight such that $f(x) = 1$ and substitute in $g(x) = 0$, until there are same number of equations as unknowns.*
5. *Solve the linear system. If there is no solution, output no annihilator of degree d, but if there is a solution then it is not clear whether there is an annihilator of degree d or not.*

Next we present the deterministic version of the original probabilistic algorithm [123, Algorithm 2].

Algorithm 5

1. *Initialize weight $w = 0$.*
2. *For all x's of weight w with $f(x) = 1$, substitute each x in $g(x) = 0$ to derive a linear equation in the coefficients of g, with a single coefficient of weight w. Use this equation to express this coefficient iteratively by coefficients of lower weight.*
3. *If $w < d$, increment w by 1 and go to step 2.*
4. *Substitute x such that $wt(x) > d$ and $f(x) = 1$ in $g(x) = 0$ to get linear equation in the coefficient of g.*

5. *Solve the linear system. Output no annihilator of degree d iff there is no non zero solution.*

Since first three steps of both algorithms are same, we initially study the time and space complexity of both the algorithms for first three steps for a randomly chosen balanced function f. In step 2, we apply x, such that $\mathsf{wt}(x) \leq d$ and $f(x) = 1$, in $g(x)$ and hence we get a linear equation in the coefficient of g such that a single coefficient of that weight is expressed as linear combination of its lower weight coefficients. Here we consider a particular w for each iteration. As f is random and balanced, one can expect that there are $\frac{1}{2}\binom{n}{w}$ many input vectors of weight w in set $supp(f)$. For each $x = (x_1, \ldots, x_n) \in supp(f)$ where $x_{i_1}, \ldots, x_{i_w}$ are 1 and others are 0 of weight w, we will get linear equation of the form

$$a_{i_1,\ldots,i_w} = a_0 + \sum_{j=1}^{w} a_{i_j} + \cdots + \sum_{\{k_1,\ldots,k_{w-1}\} \subset \{i_1,\ldots i_w\}} a_{k_1,\ldots,k_{w-1}}. \tag{7.3}$$

To store one equation we need $\sum_{i=0}^{w}\binom{n}{i}$ many memory bits (some places will be 0, some will be 1). There are $\sum_{i=0}^{w-1}\binom{w}{i}$ many coefficients in the right hand side of the Equation 7.3. As f is random, one can expect that half of them can be eliminated using the equations obtained by lower weight input support vectors. So, $\sum_{i=0}^{w}\binom{n}{i} + \frac{1}{2}\sum_{i=0}^{w-1}(\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j})$ order of computation is required to establish an equation. Here w varies from 0 to d and there are approximately $\frac{1}{2}\sum_{w=0}^{d}\binom{n}{w}$ many support vectors of weight less than or equal to d. Hence at the starting of step 4 the space complexity is

$$S1 = \frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}\sum_{i=0}^{w}\binom{n}{i})$$

and time complexity is

$$T1 = \frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}(\sum_{i=0}^{w}\binom{n}{i} + \frac{1}{2}\sum_{i=0}^{w-1}\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j})).$$

Now we study the time and space complexity for steps 4 and 5 in both probabilistic and deterministic version. To represent each equation for the system of equation one needs $\sum_{w=0}^{d}\binom{n}{w}$ memory bits.

First we consider the probabilistic one. For probabilistic case one has to choose approximately $\frac{1}{2}\sum_{w=0}^{d}\binom{n}{w}$ many support input vectors of weight greater than d. Hence each linear equation obtained from these vectors has at least $\sum_{i=0}^{d}\binom{d+1}{i}$ many coefficients of g

and half of them can be eliminated using the equations obtained in previous steps. So, to get each equation one needs at least $\sum_{w=0}^{d}\binom{n}{w}+\frac{1}{2}\sum_{i=0}^{d}(\binom{d+1}{i}\sum_{j=0}^{i-1}\binom{n}{j})$ computations. Hence the space complexity during 4th step is $SP2 \geq \frac{1}{2}(\sum_{w=0}^{d}\binom{n}{w})^2$ and time complexity is $TP2 \geq \frac{1}{2}\sum_{w=0}^{d}\binom{n}{w}(\sum_{w=0}^{d}\binom{n}{w}+\frac{1}{2}\sum_{i=0}^{d}(\binom{d+1}{i}\sum_{j=0}^{i-1}\binom{n}{j}))$. Finally, to generate system of homogeneous linear equations one requires

$$SP = S1 + SP2 \geq \frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}\sum_{i=0}^{w}\binom{n}{i}) + \frac{1}{2}(\sum_{w=0}^{d}\binom{n}{w})^2$$

memory bits and

$$TP = T1+TP2 \geq \frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}(\sum_{i=0}^{w}\binom{n}{i}+\frac{1}{2}\sum_{i=0}^{w-1}\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j}))+\frac{1}{2}\sum_{w=0}^{d}\binom{n}{w}(\sum_{w=0}^{d}\binom{n}{w}+$$
$$\frac{1}{2}\sum_{i=0}^{d}(\binom{d+1}{i}\sum_{j=0}^{i-1}\binom{n}{j}))$$

computations. Then in step 5, we have to solve $\frac{1}{2}\sum_{w=0}^{d}\binom{n}{w}$ many linear equations with same number of variables. To solve this system one needs $TP3 = (\frac{1}{2}\sum_{w=0}^{d}\binom{n}{w})^3$ computations using the Gaussian elimination technique.

Now we study space and time complexity for deterministic one. Since f is balanced, there are approximately $2^{n-1} - \frac{1}{2}\sum_{w=0}^{d}\binom{n}{w} = \frac{1}{2}\sum_{w=d+1}^{n}\binom{n}{w}$ many support vectors having weight greater than d and these many are considered to find out equations. Hence each linear equation obtained from these vectors of weight $w > d$ contains $\sum_{i=0}^{d}\binom{w}{i}$ many coefficients of g and half of them can be eliminated using the equations obtained in steps 1, 2 and 3. To get this equation one needs $\sum_{i=0}^{d}\binom{n}{i}+\frac{1}{4}\sum_{i=0}^{d}(\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j})$ computations. Hence the total space complexity during 4th step is $SD2 = \frac{1}{4}\sum_{w=d+1}^{n}\binom{n}{w}\sum_{w=0}^{d}\binom{n}{d})$ and time complexity is $TD2 = \frac{1}{2}\sum_{w=d+1}^{n}\binom{n}{w}(\sum_{i=0}^{d}\binom{n}{i}+\frac{1}{4}\sum_{i=0}^{d}(\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j}))$. Finally, to generate homogeneous linear equations one needs

$$SD = S1 + SD2 = \frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}\sum_{i=0}^{w}\binom{n}{i}) + \frac{1}{4}\sum_{w=d+1}^{n}\binom{n}{w}\sum_{w=0}^{d}\binom{n}{d})$$

memory bits and

$$TD = T1+TD2 = \frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}(\sum_{i=0}^{w}\binom{n}{i}+\frac{1}{2}\sum_{i=0}^{w-1}\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j}))+\frac{1}{2}\sum_{w=d+1}^{n}\binom{n}{w}(\sum_{i=0}^{d}\binom{n}{i}+$$
$$\frac{1}{4}\sum_{i=0}^{d}(\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j})$$

computations. Further, in step 5, we have to solve $\frac{1}{2}\sum_{w=d+1}^{n}\binom{n}{w}$ many linear equations with $\frac{1}{2}\sum_{w=0}^{d}\binom{n}{w}$ number of variables. To solve this system one needs $TD3 = (\frac{1}{2}\sum_{w=d+1}^{n}\binom{n}{w})^3$

computations.

The system of equations generated by our strategy as well as Meier et al [123] algorithms are same. So, it takes same complexities to solve them. Only difference is during generation of the system of equations. In the following table we show the complexities for both algorithms for generating the system of equations.

	Space	Time
Meier's algorithm	$\frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}\sum_{i=0}^{w}\binom{n}{i}) + \frac{1}{2}(\sum_{w=0}^{d}\binom{n}{w})^2$	$\frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}(\sum_{i=0}^{w}\binom{n}{i} + \frac{1}{2}\sum_{i=0}^{w-1}\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j}))$ $+\frac{1}{2}\sum_{w=0}^{d}\binom{n}{w}(\sum_{w=0}^{d}\binom{n}{w} + \frac{1}{2}\sum_{i=0}^{d}(\binom{d+1}{i}\sum_{j=0}^{i-1}\binom{n}{j}))$
Our algorithm	$\frac{1}{4}(\sum_{w=0}^{d}\binom{n}{w})^2$	$\frac{1}{4}(\sum_{w=0}^{d}\binom{n}{w})^2$

Table 7.1: Time and Space complexity comparison of Probabilistic algorithms to generate equations.

	Space	Time
Meier's algorithm	$\frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}\sum_{i=0}^{w}\binom{n}{i}) +$ $\frac{1}{4}\sum_{w=d+1}^{n}\binom{n}{w}\sum_{w=0}^{d}\binom{n}{d})$	$\frac{1}{2}\sum_{w=0}^{d}(\binom{n}{w}(\sum_{i=0}^{w}\binom{n}{i} + \frac{1}{2}\sum_{i=0}^{w-1}\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j}))$ $+\frac{1}{2}\sum_{w=d+1}^{n}\binom{n}{w}(\sum_{i=0}^{d}\binom{n}{i} + \frac{1}{4}\sum_{i=0}^{d}(\binom{w}{i}\sum_{j=0}^{i-1}\binom{n}{j}))$
Our algorithm	$\frac{1}{4}\sum_{w=d+1}^{n}\binom{n}{w}\sum_{w=0}^{d}\binom{n}{w}$	$\frac{1}{4}\sum_{w=d+1}^{n}\binom{n}{w}\sum_{w=0}^{d}\binom{n}{w}$

Table 7.2: Time and Space complexity comparison of Deterministic algorithms to generate equations.

7.3 Further Reduction in Matrix Size Applying Linear Transformation over the Input Variables of the Function

To check for the annihilators, we need to compute the rank of the matrix UA^r. Following Theorem 20, it is clear that the size of the matrix UA^r will decrease if μ_f increases and ν_f decreases. Let B be an $n \times n$ nonsingular binary matrix and b be an n-bit vector. The function $f(x)$ has an annihilator at degree d iff $f(Bx + b)$ has an annihilator at degree d. Thus one will try to get the affine transformation on the input variables of $f(x)$ to get $h(x) = f(Bx+b)$ such that $|\{x \mid h(x) = 1, \mathsf{wt}(x) \leq d\}|$ is maximized. This is because in this case μ_h will be maximized and ν_h will be minimized and hence the dimension of the matrix

UA^r, i.e., $\nu_f \times (\sum_{i=0}^{d} \binom{n}{i} - \mu_f)$ will be minimized. This will indeed decrease the complexity at the construction step (discussed in the previous section). More importantly, it will decrease the complexity to solve the system of homogeneous linear equations.

See the following example that explains the efficiency for a 5-variable function.

Example 8 *We present an example for this purpose. Consider the* 5*-variable Boolean function* $f(x_1, \ldots, x_5) = x_1 + \phi_4(x_2, x_3, x_4, x_5)$ *where* ϕ_4 *is constructed using the method presented in Chapter 4 such that neither* f *nor* $1 + f$ *has an annihilator at a degree less than* 3*. The standard truth table representation of the function is* 01010110010101100101011001101001*, i.e., the outputs are corresponding to the inputs which are of increasing value. One can check that* $|\{x \in \mathbb{F}_2^5 \mid f(x) = 1 \ \& \ \mathsf{wt}(x) < 3\}| = 6$*. Thus, following our strategy one has to check the rank of a* 10×10 *matrix. Now if we consider the function* $h(x) = f(Bx + b)$ *such that*

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \text{ and } b = \{1, 1, 0, 0, 1\},$$

then $|\{x \in \{0,1\}^5 \mid h(x) = 1 \ \& \ \mathsf{wt}(x) < 3\}| = 16$ *and one can immediately conclude (from the results in Chapter 5) that neither* h *nor* $1 + h$ *has an annihilator of degree less than* 3*. This is an example where after finding the affine transformation there is even no need for the solution step at all. For the function* f*, here* $h(x) = f(Bx + b)$ *such that* $|\{x \mid h(x) = 1, \mathsf{wt}(x) \leq d\}|$ *is maximized.*

We also present an example for a sub optimal case. In this case we consider

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}, \text{ and } b = \{0, 0, 0, 0, 0\},$$

then $|\{x \in \mathbb{F}_2^5 \mid h(x) = 1 \ \& \ \mathsf{wt}(x) < 3\}| = 14$*. Thus the dimension of the matrix* UA^r *becomes* 2×2 *as* $\nu_f = 2$ *and* $\sum_{i=0}^{d} \binom{n}{i} - \mu_f = 2$*. Thus one needs to check the rank of a* 2×2 *matrix instead of checking the rank of a* 10×10 *matrix.*

Now the question is how to find such an affine transformation (for the optimal or even for sub optimal cases) efficiently.

For exhaustive search to get the optimal affine transform one needs to check $f(Bx+b)$ for all $n \times n$ nonsingular binary matrices B and n bit vectors b. Since there are $\prod_{i=0}^{n-1}(2^n - 2^i)$ many nonsingular binary matrices and 2^n many n bit vectors, one needs to check $2^n \prod_{i=0}^{n-1}(2^n - 2^i)$ many cases for an exhaustive search. As weight of the input vectors are invariant under permutation of the arguments, checking for only one nonsingular matrix from the set of all nonsingular matrices whose rows are equivalent under certain permutation will suffice. Hence the exact number of search options is $\frac{1}{n!}2^n \prod_{i=0}^{n-1}(2^n - 2^i)$. One can check for $n \times n$ nonsingular binary matrices B where $row_i < row_j$ for $i < j$ (row_i is the decimal value of binary pattern of ith row). It is clear that the search is infeasible for $n \geq 8$.

Now we present a heuristic towards this. Our aim is to find out an affine transformation $h(x)$ of $f(x)$, i.e., $h(x) = f(Bx+b)$, which maximizes the value of μ_h. This means the weight of the most of the input vectors having weight less than or equal to d should be in $supp(h)$. So we attempt to get an affine transformation for a Boolean function f such that the transformation increases the probability that an input vector, having output 1, will be translated to a low weight input vector.

Consider $h(Vx+v) = f(x)$, where V is an $n \times n$ binary nonsingular matrix and $v = (v_1, v_2, \ldots, v_n) \in \mathbb{F}_2^n$. Suppose $r_1, r_2, \ldots, r_n \in \mathbb{F}_2^n$ are the row vectors of the transformation V. By $Vx + v = y$ we mean $Vx^{tr} + v = y^{tr}$, where $x = (x_1, x_2, \ldots, x_n), y = (y_1, y_2, \ldots, y_n) \in \mathbb{F}_2^n$. Given an x, we find a y by this transformation and then $h(y)$ is assigned to the value of $f(x)$. If $f(x) = 1$, we like that the corresponding $y = Vx + v$ should be of low weight. The chance of $(y_1, y_2, \ldots, y_n)$ getting low weight increases if the probability of $y_i = 0, 1 \leq i \leq n$ is increased. That means the probability of $r_i \cdot (x_1, x_2, \ldots, x_n) + v_i = 0$ for $1 \leq i \leq n$ needs to be increased. Hence we will like to choose a linearly independent set $r_i \in \mathbb{F}_2^n$, $1 \leq i \leq n$ and $v \in \mathbb{F}_2^n$ such that the probability $r_i \cdot (x_1, x_2, \ldots, x_n) + v_i = 0$, $1 \leq i \leq n$ is high when $(x_1, x_2, \ldots, x_n) \in supp(f)$. Since we use the relations $h(Vx+v) = f(x)$ and $h(x) = f(Bx+b)$, that means $B = V^{-1}$ and $b = V^{-1}v$.

The heuristic is presented below. By $bin[i]$ we denote the n-bit binary representation of the integer i.

Heuristic 1

1. *loop* $= 0$; *max* $= |\{x \mid f(x) = 1, \mathsf{wt}(x) \leq d\}|$;
2. *For* $(i = 1; i < 2^n; i++)$ {
   (a) $t = |\{x = (x_1, x_2, \ldots, x_n) \in supp(f) \mid bin[i] \cdot x = 0\}|$

(b) if $t \geq \frac{\mathsf{wt}(f)}{2}, val[i] = t$ *and* $a_i = 0$ *else* $val[i] = \mathsf{wt}(f) - t$ *and* $a_i = 1$.

}

3. *Arrange the triplets* $(bin[i], a_i, val[i])$ *in descending order of* $val[i]$.

4. *Choose suitable* n *many triplets* (r_j, v_j, k_j) *for* $1 \leq j \leq n$ *such that* r_j*s are linearly independent and* k_j*'s are high.*

5. *Construct the nonsingular matrix* V *taking* r_j, $1 \leq j \leq n$ *as* j*-th row and* $v = (v_1, v_2, \ldots, v_n)$.

6. $B = V^{-1}$, $b = V^{-1}v$.

7. *if* $max < |\{x \mid f(Bx + b) = 1, \mathsf{wt}(x) \leq d\}|$ *replace* $f(x)$ *by* $f(Bx + b)$ *and update max by* $|\{x \mid f(Bx + b) = 1, \mathsf{wt}(x) \leq d\}|$.

8. *Increment loop by* 1*; while* $(loop < maxval)$

   Go to step 2.

The time complexity of this heuristic is $(maxval \times n2^{2n})$. See the following example, where we trace Heuristic 1 for the 5-variable function f given in Example 8.

Example 9 *We have* $f = 01010110010101100101011001101001$ *and check that* $|\{x \in \mathbb{F}_2^5 \mid f(x) = 1 \ \& \ \mathsf{wt}(x) \leq 2\}| = 6$. *In step 2, we get* $(val[i], a_i)$ *for* $1 \leq i \leq 31$ *as* $1 : (11, 1)$, $2 : (8, 1)$, $3 : (11, 1)$, $4 : (8, 1)$, $5 : (11, 1)$, $6 : (8, 1)$, $7 : (9, 0)$, $8 : (8, 1)$, $9 : (9, 1)$, $10 : (8, 1)$, $11 : (9, 1)$, $12 : (8, 1)$, $13 : (9, 1)$, $14 : (8, 1)$, $15 : (11, 0)$, $16 : (8, 1)$, $17 : (9, 1)$, $18 : (8, 1)$, $19 : (9, 1)$, $20 : (8, 1)$, $21 : (9, 1)$, $22 : (8, 1)$, $23 : (11, 0)$, $24 : (8, 1)$, $25 : (9, 0)$, $26 : (8, 1)$, $27 : (9, 0)$, $28 : (8, 1)$, $29 : (9, 0)$, $30 : (8, 1)$, $31 : (11, 1)$. *Then after ordering according the value of* $val[i]$*, we choose the row of matrix* V *as the* 5*-bit binary expansion of* $1, 3, 5, 15$ *and* 7 *with frequency values of* 0*'s as* $11, 11, 11, 11, 9$ *respectively and* $v = (a_1, a_3, a_5, a_{15}, a_7) = (1, 1, 1, 1, 0)$. *Here the matrix* V *is a nonsingular matrix. The new function is* $g = f(Bx+b)$, *where* $B = V^{-1}$, $b = V^{-1}v$ *and one can check that* $|\{x \in \mathbb{F}_2^5 \mid g(x) = 1 \ \& \ \mathsf{wt}(x) \leq 2\}| = 16$.

Experiments with this heuristic on different Boolean functions provide very positive results. First of all we have considered the functions which are random affine transformations $g(x)$ of the function from Chapter 5, $f_s(x) = 1$ for $\mathsf{wt}(x) \leq \lfloor \frac{n-1}{2} \rfloor$ and $f_s(x) = 0$ for $\mathsf{wt}(x) \geq \lfloor \frac{n+1}{2} \rfloor$, which has no annihilator having degree less than or equal to $\lfloor \frac{n-1}{2} \rfloor$. This

experimentation has been done for $n = 5$ to 16. For all the cases running Heuristic 1 on $g(x)$ we could go back to $f_s(x)$. Then we have randomly changed $2^{\zeta n}$ bits on the upper half of $f_s(x)$ ($0.5 \leq \zeta \leq 0.8$ at steps of 0.1) to get $f'_s(x)$ and then put random transformations on $f'_s(x)$ to get $g(x)$. Running Heuristic 1, we could also go back to $f'_s(x)$ easily. For experiments we have taken $maxval = 20$.

The important issue is exactly when this matrix size is asymptotically reduced than the trivial matrix size $\mathsf{wt}(f) \times \sum_{i=0}^{d} \binom{n}{i}$ if one writes down the equations by looking at the truth table of the function only. This happens only when μ_f is very close to $\sum_{i=0}^{d} \binom{n}{i}$. Let $\sum_{i=0}^{d} \binom{n}{i} - \mu_f \leq 2^{\zeta n}$, where ζ is a constant such that $0 < \zeta < 1$. In that case the matrix size will be less than or equal to $(\mathsf{wt}(f) + 2^{\zeta n} - \sum_{i=0}^{d} \binom{n}{i}) \times 2^{\zeta n}$. When $d = \lfloor \frac{n}{2} \rfloor$ and n odd, $\sum_{i=0}^{d} \binom{n}{i} = 2^{n-1}$. Thus for a balanced function, the size of the matrix becomes as low as $2^{\zeta n} \times 2^{\zeta n}$. We summarize the result as follows.

Theorem 21 *Predetermine a constant ζ, such that $0 < \zeta < 1$. Consider any Boolean function $f(x) \in \mathcal{B}_n$ for which there exist a nonsingular binary matrix B and an n-bit vector b such that $\sum_{i=0}^{d} \binom{n}{i} - |\{x \mid f(Bx + b) = 1, \mathsf{wt}(x) \leq d\}| \leq 2^{\zeta n}$. If B and b are known, then the size of the matrix UA^r will be less than or equal to $(\mathsf{wt}(f) + 2^{\zeta n} - \sum_{i=0}^{d} \binom{n}{i}) \times 2^{\zeta n}$ which is asymptotically reduced in size than $\mathsf{wt}(f) \times \sum_{i=0}^{d} \binom{n}{i}$.*

That B, b can be known is quite likely from the experimental results available running Heuristic 1.

Next we have run our heuristics on randomly chosen balanced functions. The number of inputs up to weight d for a Boolean function is $\sum_{i=0}^{d} \binom{n}{i}$. Thus for a randomly chosen balanced function, it is expected that there will be $\frac{1}{2} \sum_{i=0}^{d} \binom{n}{i}$ many inputs up to weight d for which the outputs are 1. Below we present the improvement (on an average of 100 experiments in each case) we got after running Heuristic 1 with $maxval = 20$ for $n = 12$ to 16.

n	12			13			14			15			16		
d	3	4	5	4	5	6	4	5	6	5	6	7	5	6	7
$\sum_{i=0}^{d} \binom{n}{i}$	299	794	1586	1093	2380	4096	1471	3473	6476	4944	9949	16384	6885	14893	26333
$\lceil \frac{1}{2} \sum_{i=0}^{d} \binom{n}{i} \rceil$	149	397	793	541	1190	2048	735	1736	3238	2472	4974	8192	3442	7446	13166
Heuristic value	228	535	964	717	1438	2322	957	2051	3648	2917	5525	8811	3995	8194	14114

Table 7.3: Efficiency of Heuristic 1 on random balanced functions.

It should be noted that after running our heuristic on random balanced functions, the improvement is not extremely significant. There are improvements as we find that the the values are significantly more than $\frac{1}{2}\sum_{i=0}^{d}\binom{n}{i}$ (making our algorithm efficient), but the value is not very close to $\sum_{i=0}^{d}\binom{n}{i}$. This is not a problem with the efficiency of the heuristic, but with the inherent property of a random Boolean function that there may not be an affine transformation at all on $f(x)$ such that $|\{x \mid f(Bx+b) = 1, \mathsf{wt}(x) \leq d\}|$ is very high. In fact we can show that for highly nonlinear functions $f(x)$, the increment from $|\{x \mid f(x) = 1, \mathsf{wt}(x) \leq d\}|$ to $|\{x \mid f(Bx+b) = 1, \mathsf{wt}(x) \leq d\}|$ may not be significant for any B, b. The reason for this is as follows.

Proposition 22 *Let $f \in \mathcal{B}_n$ be a balanced function (n odd) having nonlinearity $\mathsf{nl}(f) = 2^{n-1} - 2^{\frac{n-1}{2}}$. Then for any nonsingular $n \times n$ matrix B and any n-bit vector b, $2^{n-1} - |\{x \mid f(Bx+b) = 1, \mathsf{wt}(x) \leq \frac{n-1}{2}\}| \geq \frac{1}{2}\binom{n-1}{\frac{n-1}{2}} - 2^{\frac{n-1}{2}-1}$.*

Proof : Let $f \in \mathcal{B}_n$ be a balanced function (n odd) having nonlinearity $\mathsf{nl}(f) = 2^{n-1} - 2^{\frac{n-1}{2}}$. Let $g \in \mathcal{B}_n$ be the function such that $g(x) = 1$ iff $\mathsf{wt}(x) \leq \frac{n-1}{2}$. By Theorem 13 in Chapter 5, $\mathsf{nl}(g) = 2^{n-1} - \binom{n-1}{\frac{n-1}{2}}$. Now we like to find out a function $h(x) = f(Bx+b)$ such that $|\{x \mid h(x) = 1, \mathsf{wt}(x) \leq \frac{n-1}{2}\}|$ is high. Consider the value $T = |supp(g) \cap supp(h)|$, i.e., $T = |\{x \mid h(x) = 1 \ \& \ \mathsf{wt}(x) \leq \frac{n-1}{2}\}|$. Without loss of generality consider $T \geq 2^{n-2}$. Hence, $d(h,g) = 2(2^{n-1} - T) = 2^n - 2T$. Now, $\mathsf{nl}(f) = \mathsf{nl}(h) \leq \mathsf{nl}(g) + d(h,g) = (2^{n-1} - \binom{n-1}{\frac{n-1}{2}}) + 2^n - 2T$. Thus, $2^{n-1} - 2^{\frac{n-1}{2}} \leq (2^{n-1} - \binom{n-1}{\frac{n-1}{2}}) + 2^n - 2T$, i.e., $2^{n-1} - T \geq \frac{1}{2}\binom{n-1}{\frac{n-1}{2}} - 2^{\frac{n-1}{2}-1}$. ■

Thus if one predetermines a ζ, then for a large n we may not satisfy the condition that $\sum_{i=0}^{\frac{n-1}{2}}\binom{n}{i} - |\{x \mid f(Bx+b) = 1, \mathsf{wt}(x) \leq d\}| \leq 2^{\zeta n}$. In this direction we present the following general result where the constraint of nonlinearity is removed.

Theorem 22 *Suppose $f \in \mathcal{B}_n$ be a randomly chosen balanced function. Then the probability to get an affine transformation such that*

$$|\{x \mid f(Bx+b) = 1, \mathsf{wt}(x) \leq \lfloor \tfrac{n-1}{2} \rfloor\}| > \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor}\binom{n}{i} - k \text{ is}$$

1. *less than* $\dfrac{(n+1)2^n \sum_{i=0}^{k-1}\binom{2^{n-1}}{i}^2}{\binom{2^n}{2^{n-1}}}$ *for n odd.*

2. *less than* $\dfrac{(n+1)2^n \sum_{i=0}^{k-1}\binom{\sum_{j=0}^{\frac{n}{2}-1}\binom{n}{j}}{i}\binom{2^n - \sum_{j=0}^{\frac{n}{2}-1}\binom{n}{j}}{i+\frac{1}{2}\binom{n}{\frac{n}{2}}}}{\binom{2^n}{2^{n-1}}}$ *for n even.*

Proof : First we prove it for n odd. The number of balanced functions $h \in \mathcal{B}_n$ such that $|\{x \mid h(x) = 1, \mathsf{wt}(x) \leq \frac{n-1}{2}\}| > 2^{n-1} - k$ is $\sum_{i=0}^{k-1} \binom{2^{n-1}}{i}^2$ (consider the upper and lower half in the truth table of the function). So, there will be at most $\sum_{i=0}^{k-1} \binom{2^{n-1}}{i}^2$ many affinely invariant classes of such functions. Further the total number of balanced function is $\binom{2^n}{2^{n-1}}$. As there are $(2^n-1)(2^n-2^1)\ldots(2^n-2^{n-1})$ many $n \times n$ nonsingular binary matrices and 2^n vectors in $\mathbb{F}_2^n$, the total number of affinely invariant classes of balanced function is greater than or equal to $\frac{\binom{2^n}{2^{n-1}}}{2^n(2^n-1)(2^n-2^1)\ldots(2^n-2^{n-1})} > \frac{\binom{2^n}{2^{n-1}}}{(n+1)2^n}$. Hence the probability of a randomly chosen balanced function will be function type h is bounded by $\frac{(n+1)2^n \sum_{i=0}^{k-1} \binom{2^{n-1}}{i}^2}{\binom{2^n}{2^{n-1}}}$. Similarly, the case for n even can be proved. ■

If one takes $k \leq 2^{\frac{3}{4}n}$, then it can be checked easily that the probability decreases fast towards zero as n increases. Thus for a random balanced function f, the probability of getting an affine transformation (which generates the function h from f) such that $|\{x \mid f(Bx+b) = 1, \mathsf{wt}(x) \leq \lfloor \frac{n-1}{2} \rfloor\}| > \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n}{i} - 2^{\frac{3}{4}n}$ is almost improbable.

Thus when one randomly chosen balanced function is considered, using the strategy of considering the function after affine transformation, one can indeed reduce the matrix size by constant factor, but the reduction may not be significant in asymptotic terms when the annihilators at the degree of $\lfloor \frac{n-1}{2} \rfloor$ are considered for large n.

7.4 Conclusion

In this chapter we present how to reduce the matrix size which is involved in finding the annihilators of a Boolean function. Our results show that considerable reduction in the size of the matrix is achievable. We identify the classes where it provides asymptotic improvement. We also note that for randomly chosen balanced functions, the improvement is rather constant than asymptotic. The reduction in matrix size helps in running the actual annihilator finding steps by Gaussian elimination method. Though our method is less efficient in general than the recently known efficient algorithms [4, 69] to find the annihilators, this work helps in theoretically understanding the structure of the matrix involved.

Chapter 8

Conclusion and Open Problems

In this chapter we summarize the contribution in this thesis with pointers to some important open questions.

In Chapter 3, we presented a lower bound on nonlinearity of a Boolean function for certain AI values relating the Walsh spectrum values to AI. Later this bound is tightened by Lobanov [111] by noting the maximum absolute value in the Walsh spectrum. It will be of great interest to relate the distribution in the Walsh spectrum with AI as in that case one may relate AI with order of resiliency of a function. Algebraic attacks become more stronger and efficient if one gets more number of low degree linearly independent annihilators. Towards this we have presented some enumeration results on low degree linearly independent annihilators of a Boolean function. In certain cases the enumeration results are tight, but there are cases when we only present certain bounds. There is space to tighten these bounds. We have studied AI of a Boolean function in terms of AI of its sub functions. There are lot of constructions (see [150, 32] and the references in these papers) where functions of lower number of variables are concatenated to construct a function on higher number of variables. Thus, study of AI in terms of AI of its sub functions needs to be analysed further and using these ideas one may construct functions with optimal AI which are good in terms of other cryptographic properties. Also a disciplined study of existing constructions of cryptographically significant Boolean functions in terms of AI is welcome. We have studied some of them. One may also note the well referred Tarannikov and related constructions [170, 136]. We have shown that the algebraic immunity of this kind of construction is non decreasing and later [18] it has been shown that the AI of this kind of n-variable functions are $\Omega(\sqrt{n})$. In this kind of construction, in one iteration, from k-variable functions $(k+3)$-variable functions are generated and the algebraic degree is increased by 1, whereas the order of resiliency is

increased by 2. It seems (and also experimental results suggest) that the algebraic immunity may increase by 1 after constant number of such iterations and thus it may be possible to show that the AI of such an n-variable function may be $O(n)$, i.e., of the form $\frac{n}{c}$, where c is a constant.

In Chapter 4 we presented the first ever construction to generate Boolean functions having optimal AI. This construction is recursive in nature. We studied some other cryptographic properties of this functions. The nonlinearity of the functions is not good enough to use the functions directly as cryptographic primitives. However, these functions can be used in conjunction with functions having good cryptographic properties.

In Chapter 5, we presented a general theory to construct Boolean functions having optimal AI. Using the theory, we present a basic construction of symmetric functions having maximum algebraic immunity. The functions can be suitably modified to get non symmetric functions. We study the algebraic degree and nonlinearity of these symmetric functions in detail.

The most important open question with respect to Chapters 4, 5 is how to construct a Boolean function with maximum possible AI having very good nonlinearity. So far number of attempts have been made, but the solution stayed elusive. Once this problem is solved, one may attempt to include other properties like resiliency or propagation characteristics in the construction.

Apart from algebraic immunity, immunity of Boolean functions against fast algebraic attacks needs serious study. In Chapter 6 we studied the immunity of Boolean functions against certain kinds of fast algebraic attacks. We evaluated how a Boolean function with optimal AI behaves against fast algebraic attacks, i.e., in terms of $fg = h$ relationships. In this direction, we have presented some experimental and theoretical results on the functions described in the earlier chapters. Then we propose some construction ideas to generate functions having properties additional to AI to provide resistance against certain kinds of fast algebraic attacks. Once the problem related to construction of functions with very high nonlinearity and maximum possible algebraic immunity is successfully solved, one may consider these additional properties too.

In [123], initial algorithms have been proposed for checking existence of annihilators and finding annihilators at a certain degree. In this process one has to solve a system of homogeneous linear equations. For a balanced function the system contains 2^{n-1} equations where n is the number of variables of f. Solving for a $n > 15$ becomes costly in this method. In Chapter 7, we have exploited some interesting structure of the matrix involved in the

system of homogeneous linear equations. Further we propose a heuristic to provide an affine transformation on the input variables of a Boolean function that reduces the size of the matrix further. Recently, two very efficient algorithms have been proposed in [4, 69] to find the annihilators and relations required for fast algebraic attacks. Using the algorithms, functions up to 20 variables can be analysed. It seems interesting to study whether our observation on the algebraic structures of the system of homogeneous linear equations can be efficiently exploited to design a more efficient algorithm.

The subject of studying Boolean functions having resistance against algebraic and fast algebraic attacks is at an early stage. In this thesis we tried to present a disciplined study in this direction, but a lot of issues remain to be solved satisfactorily. To reiterate, the immediate question of finding Boolean functions having maximum possible algebraic immunity and very good nonlinearity stays open at the time of completion of this thesis.

Bibliography

[1] S. Alhinai, L. M. Batten, B. Colbert and K. Wong. Algebraic Attacks on Clock-Controlled Stream Ciphers. In *Australasian Conference on Information Security and Privacy, ACISP 2006*. To be published in Lecture Notes in Computer Science. Springer-verlag, 2006.

[2] F. Armknecht. On the Existence of low-degree Equations for Algebraic Attacks. Cryptology ePrint Archive: report 2004/185, http://eprint.iacr.org/2004/185.

[3] F. Armknecht. Improving Fast Algebraic Attacks. In *Fast Software Encryption, FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 65–82. Springer-Verlag, 2004.

[4] F. Armknecht, C. Carlet, P. Gaborit, S. Kuenzli, W. Meier and O. Ruatta. Efficient computation of algebraic immunity for algebraic and fast algebraic attacks. In *Advances in Cryptology - Eurocrypt 2006*, number 4004 in Lecture Notes in Computer Science. Springer-Verlag, 2006.

[5] F. Armknecht and M. Krause. Algebraic attacks on combiners with memory. In *Advances in Cryptology - Crypto 2003*, number 2729 in Lecture Notes in Computer Science, pages 162–175. Springer-Verlag, 2003.

[6] G. Ars and J. Faugére. Algebraic Immunities of functions over finite fields. Technical report, INRIA, France, 2005.

[7] L. M. Batten. Algebraic Attacks over GF(q). In *Progress in Cryptology - Indocrypt 2004*, number 3348 in Lecture Notes in Computer Science, pages 84–91. Springer-Verlag, 2004.

[8] E. R. Berlekamp and L. R. Welch. Weight distributions of the cosets of the $(32, 6)$ Reed-Muller code. *IEEE Transactions on Information Theory*, IT-18(1):203–207, 1972.

[9] E. Biham, R. J. Anderson and L. R. Knudsen. Serpent: A New Block Cipher Proposal. In *Fast Software Encryption, FSE 1998*, number 1372 in Lecture Notes in Computer Science, pages 222–238. Springer-Verlag, 1998.

[10] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. In *Advances in Cryptology - Crypto 1990*, number 537 in Lecture Notes in Computer Science, pages 2–21. Springer-Verlag, 1991.

[11] A. Biryukov and C. D. Cannière. Block Ciphers and Systems of Quadratic Equations. In *Fast Software Encryption, FSE 2003*, number 2887 in Lecture Notes in Computer Science, pages 274–289. Springer-Verlag, 2003.

[12] A. Biryukov and A. Shamir. Cryptanalytic Time/Memory/Data tradeoffs for stream ciphers. In *Advances in Cryptology, Asiacrypt 2000*, number 1976 in Lecture Notes in Computer Science, pages 1–13. Springer-Verlag, 2000.

[13] Bluetooth SIG, *Specification of the Bluetooth system*, Version 1.1, February 22, 2001. Available at http://www.bluetooth.com.

[14] L. Blum, M. Blum, and M. Shub. A simple unpredictable random number generator. *SIAM Journal on Computing*, 15:364–383, 1986.

[15] M. Boesgaard, M. Vesterager, T. Pedersen, J. Christiansen and O. Scavenius. Rabbit: A New High-Performance Stream Cipher. In *Fast Software Encryption, FSE 2003*, number 2887 in Lecture Notes in Computer Science, pages 307–329. Springer-Verlag, 2003.

[16] A. Botev. On algebraic immunity of some recursively given sequence of correlation immune functions. In Proceedings of *XV international workshop on Synthesis and complexity of control systems*, Novosibirsk, October 18-23, 2004, pages 8-12 (in Russian).

[17] A. Botev. On algebraic immunity of new constructions of filters with high nonlinearity. In Proceedings of *VI international conference on Discrete models in the theory of control systems*, Moscow, December 7-11, 2004, pages 227-230 (in Russian).

[18] A. Botev and Y. Tarannikov. Lower bounds on algebraic immunity for recursive constructions of nonlinear filters. Preprint 2004.

[19] A. Braeken, J. Lano, N. Menetens, B. Preneel and I. Verbauwhede. SFINKS: A synchronous stream cipher for restricted hardware environments. In SKEW- Symmetric key Encryption Workshop, 2005.

[20] A. Braeken, J. Lano and B. Praneel. Evaluating the Resistance of Stream Ciphers with Linear Feedback Against Fast Algebraic Attacks. In *Australasian Conference on Information Security and Privacy, ACISP 2006*. To be published in Lecture Notes in Computer Science. Springer-verlag, 2006. An earlier version is available in Eprint on ECRYPT, 2005.

[21] A. Braeken and B. Praneel. Probabilistic algebraic attacks. In *10th IMA international conference on cryptography and coding, 2005*, number 3796 in Lecture Notes in Computer Science, pages 290–303. Springer-Verlag, 2005.

[22] A. Braeken and B. Praneel. On the Algebraic Immunity of Symmetric Boolean Functions. In *Indocrypt 2005*, number 3797 in Lecture Notes in Computer Science, pages 35–48. Springer Verlag, 2005. An earlier version is available at Cryptology ePrint Archive: report 2005/245, http://eprint.iacr.org/2005/245.

[23] B. Buchberger. Gröbner bases: an algorithmic method in polynomial ideal theory. In *Multidimensional Systems Theory*, edited by N. K. Bose, D. Reidel Publishing Company, Dordrecht, 184–232, 1985.

[24] P. Camion, C. Carlet, P. Charpin, and N. Sendrier. On Correlation-Immune Functions. In *Advances in Cryptology – Crypto 1991*, number 576 in Lecture Notes in Computer Science, pages 86–100. Springer-Verlag, 1992.

[25] A. Canteaut. Ongoing Research Areas in Symmetric Cryptography. Technical report, INRIA, France, 2005.

[26] A. Canteaut. Open problems related to algebraic attacks on stream ciphers. In *Workshop on Coding and Cryptography, WCC 2005*, pages 1–10, invited talk.

[27] A. Canteaut, C. Carlet, P. Charpin and C. Fontaine. Propagation Characteristics and Correlation-Immunity of Highly Nonlinear Boolean Functions. In *Advances in Cryptology – Eurocrypt 2000*, number 1807 in Lecture Notes in Computer Science, pages 507–522. Springer Verlag, 2000.

[28] A. Canteaut and M. Trabbia. Improved Fast Correlation Attacks Using Parity-Check Equations of Weight 4 and 5. In *Advances in Cryptology – Eurocrypt 2000*, number 1807 in Lecture Notes in Computer Science, pages 573–588. Springer Verlag, 2000.

[29] A. Canteaut and M. Videau. Degree of Composition of Highly Nonlinear Functions and Applications to Higher Order Differential Cryptanalysis. In *Advances in Cryptology – Eurocrypt 2002*, number 2332 in Lecture Notes in Computer Science, pages 518–533. Springer Verlag, 2002.

[30] A. Canteaut and M. Videau. Symmetric Boolean functions. *IEEE Transactions on Information Theory*, IT-51(8):2791–2811, 2005.

[31] C. Carlet. Two new classes of bent functions. In *Advances in Cryptology - Eurocrypt 1993*, number 765 in Lecture Notes in Computer Science, pages 77–101. Springer-Verlag, 1994.

[32] C. Carlet. A larger class of cryptographic Boolean functions via a study of the Maiorana-McFarland construction. In *Advances in Cryptology - Crypto 2002*, number 2442 in Lecture Notes in Computer Science, pages 549–564. Springer Verlag, 2002.

[33] C. Carlet. Improving the algebraic immunity of resilient and nonlinear functions and constructing bent functions. Cryptology ePrint Archive: report 2004/276, http://eprint.iacr.org/2004/276. See also the extended abstract entitled "Designing bent functions and resilient functions from known ones, without extending their number of variables" in the proceedings of ISIT 2005.

[34] C. Carlet. Concatenating indicators of flats for designing cryptographic functions. *Design, Codes and Cryptography*, 36(2):189 – 202, 2005.

[35] C. Carlet. On the Higher Order Nonlinearities of Algebraic Immune Functions. Accepted in *Advances in Cryptology - Crypto 2006*. A primary version of the paper is available at Cryptology ePrint Archive: report 2005/469, http://eprint.iacr.org/2005/469.

[36] C. Carlet, D. K. Dalai, K. C. Gupta and S. Maitra. Algebraic Immunity for Cryptographically Significant Boolean Functions: Analysis and Construction. *IEEE Transactions on Information Theory*, IT-52(7):3105 – 3121, 2006. This is a revised and extended version of [62, 63].

[37] C. Carlet, D. K. Dalai and S. Maitra. Cryptographic Properties and Structure of Boolean Functions with Full Algebraic Immunity. In *IEEE International Symposium on Information Theory, ISIT 06*, July 9 – 14, 2006, Seattle, Washington.

[38] C. Carlet and P. Gaborit. On the construction of balanced Boolean functions with a good algebraic immunity. In *First Workshop on Boolean Functions: Cryptography and Applications, BFCA 05*, March 7–9, 2005, LIFAR, University of Rouen, France.

[39] C. Carlet and P. Guillot. A characterization of bent functions. *Journal of Combinatorial Theory, Series A*, 76(2):328–335, September 1996.

[40] S. Chee, S. Lee, D. Lee, and S. H. Sung. On the correlation immune functions and their nonlinearity. In *Advances in Cryptology, Asiacrypt 1996*, number 1163 in Lecture Notes in Computer Science, pages 232–243. Springer-Verlag, 1996.

[41] J. H. Cheon and D. H. Lee. Resistance of S-Boxes against Algebraic Attacks. In *Fast Software Encryption, FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 83–94. Springer-Verlag, 2004.

[42] V. V. Chepyzhov and B. Smeets. On A Fast Correlation Attack on Certain Stream Ciphers. In *Advances in Cryptology - Eurocrypt 1991*, number 547 in Lecture Notes in Computer Science, pages 176–185. Springer Verlag, 1991.

[43] J. Y. Cho and J. Pieprzyk. Algebraic Attacks on SOBER-t32 and SOBER-128. In *Fast Software Encryption, FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 49–64. Springer Verlag, 2004.

[44] J. Clark, J. Jacob, S. Maitra and P. Stănică. Almost Boolean Functions: The Design of Boolean Functions by Spectral Inversion. *Computational Intelligence*, 20(3):450–462, 2004.

[45] L. Comtet. *Advanced combinatorics*, Reidel Publication, 1974.

[46] G. M. Constantine. Combinatorial Theory and Statistical Design. John Wiley & Sons, 1987.

[47] J. W. Cooley and J. W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier series. *Mathematics of Computation*, 19:297–301, 1990.

[48] D. Coppersmith and S. Winograd. Matrix multiplication via arithmatic programming. *Journal of Symbolic Computation*, 9(3):251-280, 1990.

[49] N. Courtois. The Security of Hidden Field Equations (HFE). In *CT-RSA 2001*, number 2020 in Lecture Notes in Computer Science, pages 266–281. Springer Verlag, 2001.

[50] N. Courtois. Higher Order Correlation Attacks, XL Algorithm and Cryptanalysis of Toyocrypt. In *International Conference in Information Security and Cryptology - ICISC 2002* , number 2587 in Lecture Notes in Computer Science, pages 182–199. Springer-Verlag, 2002.

[51] N. Courtois. Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. In *Advances in Cryptology - Crypto 2003*, number 2729 in Lecture Notes in Computer Science, pages 176–194. Springer-Verlag, 2003.

[52] N. Courtois. Algebraic Attacks on Combiners with Memory and Several Outputs. In *International Conference in Information Security and Cryptology - ICISC 2004*, number 3506 in Lecture Notes in Computer Science, pages 3–20. Springer-Verlag, 2004.

[53] N. Courtois. Cryptanalysis of SFINKS. In *International Conference in Information Security and Cryptology - ICISC 2005*, number 3935 in Lecture Notes in Computer Science, pages 261–269. Springer-Verlag, 2005. A primary version is available at Cryptology ePrint Archive: report 2005/243, http://eprint.iacr.org/2005/243.

[54] N. Courtois, B. Debraize and E. Garrido. On Exact Algebraic [Non-]Immunity of S-boxes Based on Power Functions. In *Australasian Conference on Information Security and Privacy, ACISP 2006*. To be published in Lecture Notes in Computer Science. Springer-verlag, 2006.An earlier version is available at Cryptology ePrint Archive: report 2005/203, http://eprint.iacr.org/2005/203.

[55] N. Courtois, A. Klimov, J. Patarin and A. Shamir. Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In *Advances in Cryptology - Eurocrypt 2000*, number 1807 in Lecture Notes in Computer Science, pages 392–407. Springer Verlag, 2000.

[56] N. Courtois and W. Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback. In *Advances in Cryptology - Eurocrypt 2003*, number 2656 in Lecture Notes in Computer Science, pages 345–359. Springer Verlag, 2003.

[57] N. Courtois and J. Patarin. About the XL algorithm over $GF(2)$. In *CT-RSA 2003*, number 2612 in Lecture Notes in Computer Science, pages 141–157. Springer Verlag, 2003.

[58] N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In *Advances in Cryptology - Asiacrypt 2002*, number 2501 in Lecture Notes in Computer Science, pages 267–287. Springer Verlag, 2002. Modified and extended version is available in Cryprology ePrint Archive: report 2002/044, http://eprint.iacr.org/2002/044/.

[59] D. Cox, J. Little and D. O'Shea. *Ideals, Varieties, and Algorithms.* An introduction to computational algebraic geometry and commutative algebra. Undergraduate Texts in Mathematics. Springer-Verlag, second edition, 1997.

[60] T. W. Cusick, C. Ding and A. Renvall. *Stream Ciphers and Number Theory.* North-Holland Mathematical library, volume 66, Elsevier, 2004.

[61] T. W. Cusick and P. Stănică. Fast Evaluation, Weights and Nonlinearity of Rotation-Symmetric Functions. *Discrete Mathematics* 258(1-3):289–301, 2002.

[62] D. K. Dalai, K. C. Gupta and S. Maitra. Results on Algebraic Immunity for Cryptographically Significant Boolean Functions. In *Progress in Cryptology - Indocrypt 2004*, number 3348 in Lecture Notes in Computer Science, pages 92–106. Springer Verlag, 2004.

[63] D. K. Dalai, K. C. Gupta and S. Maitra. Cryptographically Significant Boolean functions: Construction and Analysis in terms of Algebraic Immunity. In *Fast Software Encryption, FSE 2005*, number 3557 in Lecture Notes in Computer Science, pages 98–111. Springer-Verlag 2005.

[64] D. K. Dalai, K. C. Gupta and S. Maitra. Notion of Algebraic Immunity and Its evaluation Related to Fast Algebraic Attacks. In *Second International Workshop on Boolean Functions: Cryptography and Applications, BFCA 06*, March 13–15, 2006, LIFAR, University of Rouen, France. An earlier version is available at Cryptology ePrint Archive: report 2006/018, http://eprint.iacr.org/2006/018.

[65] D. K. Dalai and S. Maitra. Reducing the Number of Homogeneous Linear Equations in Finding Annihilators. Accepted in International Conference on sequences and their applications, SETA 2006, to be held in September 24 – 28, 2006 Beijing, China, to be published in number 4086 in Lecture Notes in Computer Science, Springer-verlag. An extended version is available at Cryptology ePrint Archive: report 2006/032, http://eprint.iacr.org/2006/032.

[66] D. K. Dalai, S. Maitra and S. Sarkar. Basic Theory in Construction of Boolean Functions with Maximum Possible Annihilator Immunity. *Design, Codes and Cryptography* 40(1):41–58, July 2006. A preliminary version is available at Cryptology ePrint Archive: report 2005/229, http://eprint.iacr.org/2005/229.

[67] D. K. Dalai, S. Maitra and S. Sarkar. Results on Rotation Symmetric Bent Functions. In *Second International Workshop on Boolean Functions: Cryptography and Applications, BFCA 06*, March 13–15, 2006, LIFAR, University of Rouen, France. A primary version is available at Cryptology ePrint Archive: report 2005/118, http://eprint.iacr.org/2005/118.

[68] F. Didier. A new upper bound on the block error probability after decoding over the erasure channel. Accepted in IEEE Transactions on Information Theory, 2006.

[69] F. Didier and J. Tillich. Computing the Algebraic Immunity Efficiently. In *Fast Software Encryption, FSE 2006*, to be published in Lecture Notes in Computer Science. Springer-Verlag, 2006.

[70] C. Diem. The XL-Algorithm and a Conjecture from Commutative Algebra. In *Advances in Cryptology - Asiacrypt 2004*, number 3329 in Lecture Notes in Computer Science, pages 323–337. Springer Verlag, 2004.

[71] W. Diffe and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(5):644–654, 1976.

[72] J. F. Dillon. Elementary Hadamard Difference sets. PhD Thesis, University of Maryland, 1974.

[73] C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers*. Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.

[74] H. Dobbertin. Almost Perfect Nonlinear Power Functions on $GF(2^n)$: the Niho case. *Information and Computation*, 151:57–72, 1998.

[75] H. Dobbertin. Almost Perfect Nonlinear Power Functions on $GF(2^n)$: the Welch case. *IEEE Transactions on Information Theory*, IT-45(4):1271–1275, 1999.

[76] P. Ekdahl and T. Johansson. A New Version of the Stream Cipher SNOW. In *Selected Areas in Cryptography, SAC 2002*, number 2595 in Lecture Notes in Computer Science, pages 47–61. Springer-Verlag, 2003.

[77] J. C. Faugére. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, June 1999.

[78] J. C. Faugére. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*. ACM, 2002.

[79] J. C. Faugère and A. Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. In *Advances in Cryptology - Crypto 2003*, number 2729 in Lecture Notes in Computer Science, pages 44–60. Springer-Verlag, 2003.

[80] N. Ferguson, R. Schroeppel and D. Whiting. A Simple Algebraic Representation of Rijndael. In *Selected Areas in Cryptography, SAC 2001*, number 2259 in Lecture Notes in Computer Science, pages 103–111. Springer-Verlag, 2001.

[81] N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, S. Lucks and T. Kohno. Helix: Fast Encryption and Authentication in a Single Cryptographic Primitive. In *Fast Software Encryption, FSE 2003*, number 2887 in Lecture Notes in Computer Science, pages 330–346. Springer-Verlag, 2003.

[82] E. Filiol and C. Fontaine. Highly nonlinear balanced Boolean functions with a good correlation-immunity. In *Advances in Cryptology - Eurocrypt 1998*, number 1403 in Lecture Notes in Computer Science, pages 475 – 488. Springer-Verlag, 1998.

[83] W. F. Friedman. The Index of Coincidence and Its Application in Cryptography. *Riverbank Publication*, No. 22, 1920.

[84] M. R. Garey and D. Johnson. *Computers and Intractability*. W H Freeman publisher, 1999.

[85] R. Gold. Maximal Recursive Sequences with 3-valued Recursive Cross-correlation Functions. *IEEE Transactions on Information Theory*, IT-14(1):154–156, 1968.

[86] O. Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2001.

[87] J. D. Golic. Cryptanalysis of alleged A5 Stream Cipher. In *Advances in Cryptology - EuroCrypt 1997*, number 1233 in Lecture Notes in Computer Science, pages 239–255. Springer-Verlag, 1997.

[88] G. Gong. *Sequence Analysis*. Lecture Notes for CO739x, Winter 1999. Available at website http://www.comsec.uwaterloo.ca/∼ggong.

[89] K. Gopalakrisnan, D. G. Hoffman, and D. R. Stinson. A note on a conjecture concerning symmetric resilient functions. *Information Processing Letters*, 47(3):139–143, 1993.

[90] K. C. Gupta. *Cryptographic and Combinatorial Properties of Boolean Functions and S-boxes.* PhD thesis, Indian Statistical Instituite, Kolkata, 2004.

[91] K. C. Gupta and P. Sarkar. Efficient software implementation of resilient Maiorana-McFarland S-Boxes. In *5th International Workshop on Information Security Applications, WISA 2004*, number 3325 in Lecture Notes in Computer Science, pages 317–331. Springer-Verlag, 2004.

[92] S. Halevi, D. Coppersmith and C. S. Jutla. Scream: A Software-Efficient Stream Cipher. In *Fast Software Encryption, FSE 2002*, number 2365 in Lecture Notes in Computer Science, pages 195 –209. Springer-Verlag, 2002.

[93] P. Hawkes and G. G. Rose. Rewriting Variables: The Complexity of Fast Algebraic Attacks on Stream Ciphers. In *Advances in Cryptology - Crypto 2004*, number 3152 in Lecture Notes in Computer Science, pages 390–406. Springer-Verlag, 2004.

[94] 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*, June 19–25, 2004, Black Sea Coast, Bulgaria.

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

[96] K. Hoffman and R. Kunze. *Linear Algebra.* Prentice-Hall, 1984.

[97] T. Jakobsen and L. R. Knudsen. The interpolation attack on block ciphers. In *Fast Software Encryption, FSE 1997*, number 1267 in Lecture Notes in Computer Science, pages 28–40. Springer-Verlag, 1997.

[98] T. Johansson and F. Jönsson. Fast Correlation Attacks Based on Turbo Code Techniques. In *Advances in Cryptology - Crypto 1999*, number 1666 in Lecture Notes in Computer Science, pages 181–197. Springer-Verlag, 1999.

[99] T. Johansson and F. Jönsson. Improved Fast Correlation Attacks on Stream ciphers via Convolutional Codes. In *Advances in Cryptology - Eurocrypt 1999*, number 1592 in Lecture Notes in Computer Science, pages 347–362. Springer-Verlag, 1999.

[100] T. Johansson and F. Jönsson. Fast correlation attacks through reconstruction of linear polynomials. In *Advances in Cryptology - Crypto 2000*, number 1880 in Lecture Notes in Computer Science, pages 300–315. Springer Verlag, 2000.

[101] T. Kasami. The Weight Enumerators for Several Classes of Subcodes of the Second Order Binary Reed-Muller Codes. *IEEE Transactions on Information Theory*, IT-18(4):369–394, 1971.

[102] S. Kavut, S. Maitra and M. D. Yücel. There exist Boolean functions on n (odd) variables having nonlinearity $> 2^{n-1} - 2^{\frac{n-1}{2}}$ if and only if $n > 7$. Cryptology ePrint Archive: report 2006/181, http://eprint.iacr.org/2006/181.

[103] A. Kerckhoffs. La Cryptographie Militaire. *Journal des Sciences Militaires*, 9^{th} series, IX(Jan 1883):pages 5-38, (Feb 1883):pages 161-191.

[104] A. Kipnis and A. Shamir. Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. In *Advances in Cryptology - Crypto 1999*, number 1666 in Lecture Notes in Computer Science, pages 19–30. Springer-Verlag, 1999.

[105] D. E. Knuth. *The Art of Computer Programming*, volume 2. Addison Wesley, 1969.

[106] I. Krasikov. On integral zeros of Krawtchouk polynomials. *Journal of Combinatorial Theory, Series A*, 74:71–99, 1996.

[107] M. Krause. BDD-Based Cryptanalysis of Key stream Generators. In *Advances in Cryptology – Eurocrypt 2002*, number 2332 in Lecture Notes in Computer Science, pages 222–237. Springer-Verlag, 2002.

[108] X. Lai and J. L. Massey. A Proposal for a New Block Encryption Standard. In *Advances in Cryptology – Eurocrypt 1990*, number 473 in Lecture Notes in Computer Science, pages 389–404. Springer-Verlag, 1991.

[109] D. H. Lee, J. Kim, J. Hong, J. W. Han and D. Moon. Algebraic Attacks on Summation Generators. In *Fast Software Encryption, FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 34–48. Springer Verlag, 2004.

[110] R. Lidl and H. Neiderreiter. *Finite Fields*. Second edition, Cambridge University Press, 1997.

[111] M. Lobanov. Tight bound between nonlinearity and algebraic immunity. Cryptology ePrint Archive: report 2005/441, http://eprint.iacr.org/2005/441.

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

[113] S. Maitra. *Boolean Functions with Important Cryptographic Properties*. PhD thesis, Indian Statistical Instituite, Kolkata, 2000.

[114] S. Maitra and E. Pasalic. Further Constructions of Resilient Boolean Functions with very High Nonlinearity. *IEEE Transactions on Information Theory*, IT-48(7):1825–1834, 2002.

[115] S. Maitra and P. Sarkar. Highly nonlinear resilient functions optimizing Siegenthaler's inequality. In *Advances in Cryptology – Crypto 1999*, number 1666 in Lecture Notes in Computer Science, pages 198–215. Springer Verlag, 1999.

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

[117] S. Maitra and P. Sarkar. Maximum Nonlinearity of Symmetric Boolean Functions on Odd Number of Variables. *IEEE Transactions on Information Theory*, IT-48(9):2626–2630, 2002.

[118] J. L. Massey. Shift-register Synthesis and BCH Decoding. *IEEE Transactions on Information Theory*, IT-15:122–127, 1969.

[119] M. Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology – Eurocrypt 1993*, number 765 in Lecture Notes in Computer Science, pages 386–397. Springer-Verlag, 1994.

[120] A. Maximov. Classes of Plateaued Rotation Symmetric Boolean Functions under Transformation of Walsh Spectra. In *Workshop on Coding and Cryptography, WCC 2005*, pages 325–334. A detailed version is available at Cryptology ePrint Archive: report 2004/354, http://eprint.iacr.org/2004/354.

[121] A. Maximov. *Some Words on Cryptanalysis of Stream Ciphers.* PhD thesis, Lund University, Lund, Sweden, 2006.

[122] 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*, March 7–9, 2005, LIFAR, University of Rouen,

France. An earlier version is available at Cryptology ePrint Archive: report 2004/144, http://eprint.iacr.org/2004/144.

[123] W. Meier, E. Pasalic and C. Carlet. Algebraic attacks and decomposition of Boolean functions. In *Advances in Cryptology - Eurocrypt 2004*, number 3027 in Lecture Notes in Computer Science, pages 474–491. Springer Verlag, 2004.

[124] W. Meier and O. Staffelbach. Fast Correlation Attacks on certain Stream ciphers. In *Journal of Cryptology*, 1(3):159–176, 1989.

[125] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001. Also available at http://www.cacr.math.uwaterloo.ca/hac.

[126] W. Mihaljevic and H. Imai. Cryptanalysis of Toyocrypt-HS1 stream cipher. In *IEICE Transactions on Fundamentals*, E85-A:66–73, January 2002.

[127] W. Millan, A. Clark, and E. Dawson. Heuristic design of cryptographically strong balanced Boolean functions. In *Advances in Cryptology – Eurocrypt 1998*, number 1403 in Lecture Notes in Computer Science, pages 489–499. Springer-Verlag, 1998.

[128] T. Moh. On the Method of XL and its inefficiency to TTM. Cryptology ePrint Archive: report 2001/47, http://eprint.iacr.org/2001/47.

[129] M. Morris Mano. *Digital Logic and Computer Design*. Prentice Hall (India), 1989.

[130] S. Murphy and M. J. B. Robshaw. Essential Algebraic Structure within the AES. In *Advances in Cryptology – Crypto 2002*, number 2442 in Lecture Notes in Computer Science, pages 1–16. Springer-Verlag, 2002.

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

[132] National Institute of Standards and Technology, 2001. http://csrc.nist.gov/Crypto Toolkit/aes/rijndael/ .

[133] National Institute of Standards and Technology, 1999. http://csrc.nist.gov/publica tions/fips/fips46-3/fips46-3.pdf .

[134] Y. Nawaz, G. Gong and K. C. Gupta. Upper Bounds on Algebraic Immunity of Power Functions. In *Fast Software Encryption, FSE 2006*, to be published in Lecture Notes in Computer Science. Springer-Verlag, 2006.

[135] E. Pasalic. Degree optimized resilient Boolean functions from Maiorana-McFarland class. In *9-th IMA conference on Cryptography and Coding, 2003*, number 2898 in Lecture Notes in Computer Science, pages 93–114. Springer-Verlag, 2003.

[136] E. Pasalic, S. Maitra, T. Johansson and P. Sarkar. New constructions of resilient and correlation immune Boolean functions achieving upper bounds on nonlinearity. In *Workshop on Coding and Cryptography - WCC 2001*, Paris, January 8–12, 2001. Electronic Notes in Discrete Mathematics, volume 6, Elsevier Science, 2001.

[137] J. Patarin. Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt '88. In *Advances in Cryptology – Crypto 1995*, number 963 in Lecture Notes in Computer Science, pages 248–261. Springer-Verlag, 1995.

[138] J. Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Assymetric Algorithms. In *Advances in Cryptology – Eurocrypt 1996*, number 1070 in Lecture Notes in Computer Science, pages 33–48. Springer-Verlag, 1996.

[139] 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*, IT-29(3):354–356, 1983.

[140] N. J. Patterson and D. H. Wiedemann. Correction to - the covering radius of the $(2^{15}, 16)$ Reed-Muller code is at least 16276. *IEEE Transactions on Information Theory*, IT-36(2):443, 1990.

[141] J. Pieprzyk and C. X. Qu. Rotation-Symmetric Functions and Fast Hashing. *Journal of Universal Computer Science* 5(1):20–31, 1999.

[142] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle. Propagation characteristics of Boolean functions. In *Advances in Cryptology - Eurocrypt 1990*, Lecture Notes in Computer Science, pages 161–173. Springer-Verlag, 1991.

[143] R. L. Rivest, M. J. B. Robshaw and Y. L. Yin. RC6 as the AES. In *AES Candidate Conference 2000*, pages 337–342.

[144] F. S. Roberts and B. Tesman. *Applied Combinatorics*, 2nd edition. Pearson, 2005.

[145] G. G. Rose and P. Hawkes. Turing: A Fast Stream Cipher. In *Fast Software Encryption, FSE 2003*, number 2887 in Lecture Notes in Computer Science, pages 290–306. Springer-Verlag, 2003.

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

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

[148] R. A. Rueppel and O. J. Staffelbach. Products of linear recurring sequences with maximum complexity. *IEEE Transactions on Information Theory*, IT-33(1):124–131, 1987.

[149] P. Sarkar. Hiji-bij-bij: A New Stream Cipher with a Self-Synchronizing Mode of Operation. In *Progress in Cryptology – Indocrypt 2003*, number 2887 in Lecture Notes in Computer Science, pages 36–51. Springer-Verlag, 2003.

[150] P. Sarkar and S. Maitra. Construction of Nonlinear Boolean Functions with Important Cryptographic Properties. *In Advances in Cryptology – Eurocrypt 2000*, number 1807 in Lecture Notes in Computer Science, pages 485–506. Springer-Verlag, 2000.

[151] P. Sarkar and S. Maitra. Nonlinearity bounds and construction of resilient Boolean functions. In *Advances in Cryptology - Crypto 2000*, number 1880 in Lecture Notes in Computer Science, pages 515–532, Springer-Verlag, 2000.

[152] P. Savicky. On the bent Boolean functions that are symmetric. *European Journal of Combinatorics*, 15:407–410, 1994.

[153] B. Schneier, J. Kelsey, D. Whiting, D. Wagner and N. Ferguson. Comments on Twofish as an AES Candidate. In *AES Candidate Conference 2000*, pages 355–356.

[154] J. Seberry and X. M. Zhang. Highly nonlinear 0-1 balanced Boolean functions satisfying strict avalanche criterion. In *Advances in Cryptology, Asiacrypt 1992*, number 718 in Lecture Notes in Computer Science, pages 145–155. Springer-Verlag, 1993.

[155] J. Seberry, X. M. Zhang and Y. Zheng. On constructions and nonlinearity of correlation immune Boolean functions. In *Advances in Cryptology – Eurocrypt 1993*, number 765 in Lecture Notes in Computer Science, pages 181–199. Springer-Verlag, 1994.

[156] J. Seberry, X. M. Zhang and Y. Zheng. Nonlinearly balanced Boolean functions and their propagation characteristics. In *Advances in Cryptology - Crypto 1993*, number 773 in Lecture Notes in Computer Science, pages 49–60. Springer-Verlag, 1994.

[157] J. Seberry, X. M. Zhang and Y. Zheng. Structures of cryptographic functions with strong avalanche characteristics. In *Advances in Cryptology, Asiacrypt 1994*, number 917 in Lecture Notes in Computer Science, pages 119–132. Springer-Verlag, 1995.

[158] J. Seberry, X. M. Zhang and Y. Zheng. Improving the Strict Avalanche Characteristics of Cryptographic Functions. In *Information Processing Letters*, 50(1):37–41, 1995.

[159] C. E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[160] A. Shimizu and S. Miyaguchi. Fast Data Encryption Algorithm FEAL. In *Advances in Cryptology – Eurocrypt 1987*, number 304 in Lecture Notes in Computer Science, pages 267–278. Springer-Verlag, 1987.

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

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

[163] L. R. Simpson, E. Dawson, J. D. Golic and W. L. Millan. LILI Keystream Generator. In *Selected Areas in Cryptography, SAC 2000*, number 2012 in Lecture Notes in Computer Science, pages 248–261. Springer-Verlag, 2000.

[164] P. Stănică and S. Maitra. Rotation Symmetric Boolean Functions – Count and Cryptographic Properties. In *R. C. Bose Centenary Symposium on Discrete Mathematics and Applications*, December 2002. Electronic Notes in Discrete Mathematics, Elsevier, volume 15.

[165] P. Stănică, S. Maitra and J. Clark. Results on Rotation Symmetric Bent and Correlation Immune Boolean Functions. In *Fast Software Encryption, FSE 2004*, number 3017 in Lecture Notes in Computer Science, pages 161–177. Springer-Verlag, 2004.

[166] A. Steel. Allan Steel's Gröbner basis timing page, 2004. http://magma.maths.usyd.edu.au/users/allan/gb.

[167] D. R. Stinson. *Cryptography, Theory and Practice*. CRC Press. First Edition 1995, Second Edition 2002.

[168] V. Strassen. Guassian Elimination is not Optimal. In *Numerische Mathematik*, 13:354–356, 1969.

[169] Stream cipher project for ECrypt, 2005. http://www.ecrypt.eu.org/stream/ .

[170] Y. V. Tarannikov. On resilient Boolean functions with maximum possible nonlinearity. In *Progress in Cryptology - Indocrypt 2000*, number 1977 in Lecture Notes in Computer Science, pages 19–30. Springer Verlag, 2000.

[171] University of Sydney. The Magma computational algebra system, 2006. http://magma.maths.usyd.edu.au.

[172] D. Watanabe, S. Furuya, H. Yoshida, K. Takaragi and B. Preneel. A New Keystream Generator MUGI. In *Fast Software Encryption, FSE 2002*, number 2365 in Lecture Notes in Computer Science, pages 179–194. Springer-Verlag, 2002.

[173] A. F. Webster and S. E. Tavares. On the design of S-boxes. In *Advances in Cryptology – Crypto 1985*, number 218 in Lecture Notes in Computer Science, pages 523–534. Springer-Verlag, 1986.

[174] D. H. Wiedemann. Solving Sparse Linear Equations over Finite Fields. *IEEE Transactions on Information Theory*, IT-32(1):54–62, 1986.

[175] G. Xiao and J. L. Massey. A Spectral Characterization of Correlation-immune Combining Functions. *IEEE Transactions on Information Theory*, IT-34(3):569–571, 1988.

[176] B. Y. Yang and J. M. Chen. Theoretical Analysis of XL over Small Fields. In *Australasian Conference on Information Security and Privacy: ACISP 2004*, number 3108 in Lecture Notes in Computer Science, pages 277–288. Springer-Verlag, 2004.

[177] E. Zenner. On the Efficiency of the Clock Control Guessing Attack. In *International Conference in Information Security and Cryptology - ICISC 2002* , number 2587 in Lecture Notes in Computer Science, pages 200–212. Springer-Verlag, 2002.

[178] E. Zenner, M. Krause and S. Lucks. Improved Cryptanalysis of the Self-Shrinking Generator. In *Australasian Conference on Information Security and Privacy, ACISP 2001*, number 2119 in Lecture Notes in Computer Science, pages 21–35. Springer-Verlag, 2001.