BOOLEAN FUNCTIONS WITH IMPORTANT CRYPTOGRAPHIC PROPERTIES

Thesis submitted to Indian Statistical Institute

BY

SUBHAMOY MAITRA
COMPUTER & STATISTICAL SERVICE CENTER
INDIAN STATISTICAL INSTITUTE

2000

(Revised)

Boolean Functions with Important Cryptographic Properties

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

by

Subhamoy Maitra
Computer & Statistical Service Center
Indian Statistical Institute
203, B. T. Road, Calcutta 700 035, INDIA
e-mail : subho@isical.ac.in

under the supervision of

Prof. Bimal Roy
Applied Statistics Unit
Indian Statistical Institute
203, B. T. Road, Calcutta 700 035, INDIA
e-mail : bimal@isical.ac.in

To those, who contribute a lot towards the development of mankind, but receive almost nothing in return.

Acknowledgements

Writing a thesis takes several years to be completed. In this period I have been enriched in numerous ways by interaction with many people. At this point, I like to thank them all.

I am most grateful to my supervisor, Prof. Bimal Roy. He has worked very closely with me in this project. Without his help, I could not even start working in this area of research. His interests have in many ways stimulated and shaped my approach to a problem. I feel really fortunate getting a chance to work with him.

I take this opportunity to express my regards to Dr. Palash Sarkar for collaborating throughout this work. For the last two years, I got the rare privilege to work with Dr. Sarkar on every working day. He has gone through a lot of existing results in this field, explained them clearly, identified number of specific problems and motivated me in proper direction. His guidance is surely the necessary and sufficient condition for completion of this thesis. I thank him once more for his invaluable cooperation.

My interest in computer science was primarily groomed by Prof. Aditya Bagchi. He has supervised my M. Tech. (Computer Science) dissertation thesis and clearly identified different research directions in related fields. His encouragement is a major source of inspiration in completion of this thesis.

I also like to thank Prof. Anish Mukhopadhyay, Prof. B. V. Rao and Dr. Sarbani Palit for carefully reading some of the portions of this thesis and providing important suggestions. My interest in theoretical computer science and mathematics was generated by my teachers Prof. K. Sikdar, Prof. Rana Barua, Prof. Bhargab B. Bhattacharya, Prof. Bhabani P. Sinha, Prof. Probal Chaudhury and Prof. Subhas C. Nandy. I take this opportunity to thank them and all my other teachers. I must also thank my friends Mr. Sounaka Mishra, Mr. Pinakpani Pal and Mr. Subhasis Pal for their support and encouragement. The computational and latex work have been carried out in Computer and Statistical Service Center and Applied Statistics Unit. I thank all the staff members of both places for maintaining a healthy atmosphere.

I also like to thank the anonymous referees for important comments on the initial version of this draft which improve both the presentation and technical quality of the thesis.

I must also express my gratitude to my family. My parents, Mr. Kalimay Maitra and Mrs. Manasi Maitra and my brother Mr. Rajarshi Maitra are always very considerate and they provided constant encouragement in completion of this dissertation. I should also thank my wife Mrs. Arpita Maitra (Banerjee) for her constant support and cooperation at the final phase of this project.

Abstract

In this thesis we concentrate on properties of cryptographically significant Boolean functions. The techniques are mainly combinatorial and provide new results on enumeration and construction of such functions.

Initially we concentrate on a particular subset of Boolean functions called the symmetric Boolean functions. A closed form expression for the Walsh transform of an arbitrary symmetric Boolean function is presented. We completely characterize the symmetric functions with maximum nonlinearity and show that the maximum nonlinearity of n-variable symmetric function can be $2^{n-1} - 2^{\lfloor \frac{n-1}{2} \rfloor}$. Moreover, new classes of symmetric balanced and symmetric correlation immune functions are considered.

We provide a randomized heuristic to construct balanced Boolean functions on n variables ($n \geq 15$ and odd) with nonlinearity strictly greater than $2^{n-1} - 2^{\frac{n-1}{2}}$. For such functions the algebraic degree is also maintained at its maximum, $n-1$. For odd $n \leq 13$, we construct balanced functions with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ and algebraic degree $n-1$. Moreover, we design optimized 1-resilient functions with currently best known nonlinearity. We also consider propagation characteristics and strict avalanche criteria. Our constructions provide balanced functions with these properties which maintain very high nonlinearity.

The set of correlation immune Boolean functions can be partitioned into several disjoint subsets with respect to the Hamming weights of their output column. It is shown that the number of n variable correlation immune functions of Hamming weight $2a+2$ is strictly greater than the number of such functions of weight $2a$ for $2a < 2^{n-1}$. We also relate the enumeration problem of correlation immune functions to the enumeration problem of balanced correlation immune functions and provide a closed form expression for the number of correlation immune functions.

We then identify some small but interesting subsets of correlation immune Boolean functions and provide some estimates on the cardinality of those subsets. We also consider a subset of correlation immune functions which satisfy one or more of a few other conditions e.g. balancedness, nondegeneracy and nonaffinity.

Moreover, we provide a new construction method using a small set of recursive operations for a large class of highly nonlinear, resilient Boolean functions with maximum possible algebraic degree. Comparisons to previous constructions show that better nonlinearity can be obtained by this method. Our technique can be used to construct functions on large number of input variables with simple hardware implementation. The architecture is programmable and can be dynamically reconfigured to compute different functions of this class.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Cryptography or the *art of secret writing* has an ancient history. Any two communicating parties (Alice and Bob) who wants to protect their exchange of information from being overheard by some unwanted eavsdropper (Oscar) has to adopt a secret protocol between themselves. This is the basic problem of cryptography. The last two decades have experienced an extremely fast development in the field of digital computers and networking. Consequently, both the commercial and military organizations are now completely dependent on the computer systems. This is the reason that the subject of cryptography has undergone a revolution. One of the key papers which marks the advent of the new era is [30]. See also [113, 76] for a more recent documentation.

Thus one of the fundamental objective of Cryptography is to enable Alice and Bob to communicate securely over an insecure channel which is observed by an opponent, Oscar. Cryptosystems, which implement such schemes, are broadly divided into two classes.

1. Private Key Cryptosystem,

2. Public Key Cryptosystem.

Here we concentrate on Private Key Cryptosystem only. See [113, 76] for references of Public Key Cryptosystems.

First we describe the basic problem. The plaintext message Π that the sender wants to transmit will be considered to be a sequence of binary digits (bits). These bits are encrypted to produce another sequence of bits called the cipher Ψ. Encryption and decryption are done using two functions $\mathcal{E}(\Pi, K)$ and $\mathcal{D}(\Psi, K)$. Both the encryption function $\mathcal{E}(\Pi, K)$ and the decryption function $\mathcal{D}(\Psi, K)$ are parameterized by the key K of the system which is chosen from a very large set of possible keys. The sender computes $\Psi = \mathcal{E}(\Pi, K)$ and sends Ψ to

the receiver. The receiver decrypts by computing $\mathcal{D}(\Psi, K) = \Pi$. The key K used by both the sender and the receiver is same.

Next we give a brief introduction on Cryptanalytic attacks which can be divided into three categories. However, in each case the basic model being used is assumed to be known, though the parameters of the system are unknown.

1. Ciphertext Only Attacks : In this case only the cipher bits are known to the adversary. From this the adversary has to recover the message and if possible the secret parameters of the model being used. Any viable cryptosystem should not be vulnerable against this class of attack.

2. Known Plaintext Attack : In this attack it is assumed that certain sets of message text, cipher text pair is known to the adversary and from these he has to find the parameters of the system. This is a stronger assumption than the previous case.

3. Chosen Plaintext Attack : This is a variation of the previous attack where the adversary is allowed to choose his own set of message texts, the rationale being that "special" messages can be used to exploit weaknesses of the system in use.

There are two approaches in designing the functions $\mathcal{E}(\Pi, K)$ and $\mathcal{D}(\Psi, K)$. First we concentrate on *stream cipher cryptography*. Here a random sequence of bits of length equal to the message length is generated. This sequence is then bitwise XOR-ed (addition modulo 2) with the message sequence and the resulting sequence is transmitted. At the receiving end, deciphering is done by generating the same random sequence and again bitwise XOR-ing the cipher bits with the random bits. The main advantage of this scheme is that if a fast random bit generator is available, then both enciphering and deciphering are very fast. The problem however is in the availability of random bit generator. If a truly random generator is used, then of course it is impossible to regenerate the sequence and hence the message cannot be recovered. To get around the problem one can generate a long sequence of pseudorandom bits and provide both the sender and the receiver with this sequence. When the sender wants to transmit, he uses only a part of the random sequence to encipher. Once used, this part is never used again. Since the communication between the sender and the receiver is synchronous, the receiver uses the correct part of the random sequence to decipher the message. Such a sequence is called an *one time pad*. One time pads have the desirable property of *perfect secrecy* in the information theoretic sense, i.e, the entropy of the message is equal to the entropy of the message given the cipher (see [113]).

Though one time pads offer perfect secrecy, in practice it is difficult to implement such a scheme, since the distribution of random bits through an insecure channel can itself become vulnerable to unwanted intrusion. So in practice one uses a pseudorandom generator at both the sender and receiver ends, set up with the same initial conditions. Thus both the sender

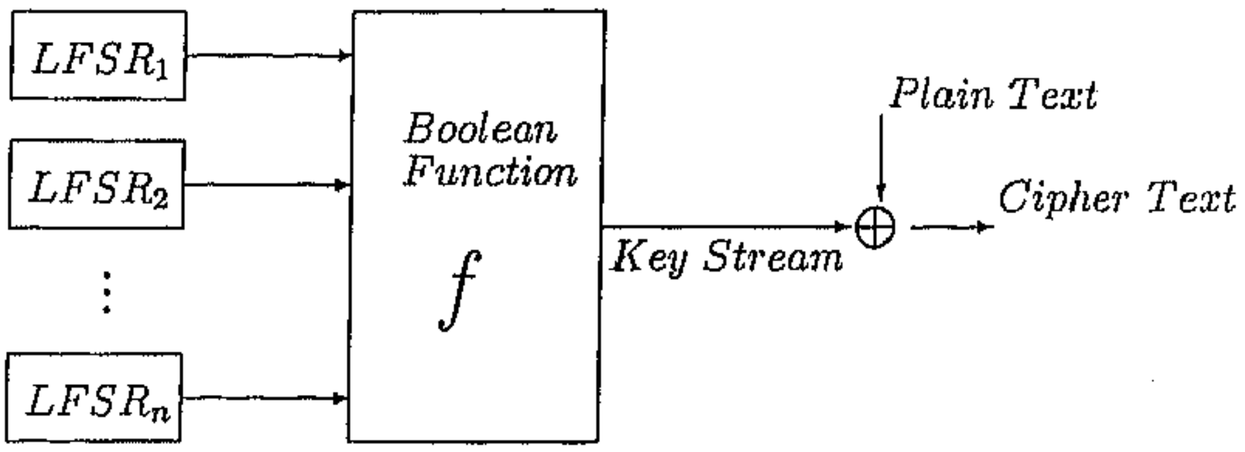


Figure 1.1: LFSR based encryption scheme

and the receiver can generate the same *random* sequence of bits. In this case the problem is to get a *good* pseudorandom generator. Many such generators have been proposed and there are statistical tests to ascertain the *pseudorandomness* of a generator (see [54]). However, for cryptographic purposes, it is important for a generator to be *secure*. Roughly, this means that given an initial sequence of generated bits $b_0, \ldots, b_k$, it should not be possible to efficiently (in polynomial time) guess the next bit with chances of success significantly greater than half. Such notions have been formalized and *cryptographically secure pseudorandom generators* have been proposed [7]. However, such systems are still not being used for practical purposes. The most popular system is based on Linear Feedback Shift Registers (LFSR) and a Boolean function as in Figure 1.1. See [38] for a detailed discussion on LFSRs. At this point it is sufficient to consider that LFSRs produce pseudorandom bit sequences with both probability of zero and probability of one equal to half. Thus, we can consider LFSRs as pseudorandom bit generators and the secret key is the initial conditions of the LFSRs, i.e. the seed of the pseudorandom bit generators. We consider the combining function as $f : GF(2^n) \rightarrow GF(2)$. The connection polynomial for each of the LFSR is usually a primitive polynomial, since this guarantees that the period of the output bit sequence is of maximal length, i.e, $2^d - 1$ for an d-bit LFSR. The function f is taken to be a nonlinear Boolean function. This is to avoid cryptanalytic attack along the lines of the *Berlekamp Massey Shift Register Synthesis Algorithm* [69, 3]. The output of f is the key sequence. This is bitwise XOR-ed with the the message bits to get the cipher bits which are then transmitted to the receiver. The receiver has an exactly same set up of the LFSR based bit generator and hence can recover the message bits by bitwise XOR-ing the cipher bits with the key bits that he generates and which is identical to the key bits generated by the sender.

For the LFSR model in Figure 1.1, if the combining Boolean function f is not properly chosen, then the system becomes susceptible to several kinds of cryptanalytic attacks. A ciphertext only cryptanalytic attack has been proposed by Siegenthaler [110]. Towards the resistivity against such divide and conquer attack, Siegenthaler himself introduced the con-

cept of correlation immunity of Boolean functions in [109]. His idea of correlation immunity was based on information theoretic measures using the concept of mutual information [43]. A characterization of information theoretic notion of correlation immunity, based on Walsh transform, is given in [42]. However, it is not sufficient to use functions with only correlation immunity, since certain types of correlation immune functions are susceptible to other kinds of attacks. For example, it is well known that the linear functions are correlation immune but not suitable for use in cryptography. We need to consider two other properties, algebraic degree [109] and nonlinearity [105]. Having a high algebraic degree ensures a high linear complexity of the produced key stream and hence better immunity against the Berlekamp-Massey shift register synthesis algorithm [69]. A high value of nonlinearity ensures that the best affine approximation [31] attack will fail. Siegenthaler, in [109], proved a fundamental inequality relating the number of variables n, order of correlation immunity m and algebraic degree d of a Boolean function: $m + d \leq n$. Moreover, if the function is balanced then $m + d \leq n - 1$. Also, a balanced m-th order correlation immune function is said to be m-resilient. A resilient Boolean function is said to be *optimized with respect to Siegenthaler's inequality* if $m + d = n - 1$.

In another approach, called *block cipher cryptography*, the message bits are divided into blocks and each block is separately enciphered and transmitted. Most of the block cipher systems use substitution boxes (S-boxes) as the nonlinear part in the scheme. These S-boxes can be viewed as a set of Boolean functions. The security of the schemes, which are based on permutations and substitutions, depends on the strength of the substitution boxes. Webster and Tavares [116] introduced the concept of strict avalanche criteria and this is an important cryptographic property for Boolean functions to be used in S-boxes. A related important property called propagation characteristics was considered by Preneel, Leekwijck, Linden, Govaerts and Vandewalle [93]. One of the most well known block cipher system is the Data Encryption Standard (DES). A detailed description of DES can be found in [113] and the structures of S-boxes in DES have been explained in [8]. Biham and Shamir [6] had proposed a differential attack on DES. Later, Matsui [71] introduced linear cryptanalysis of the Data Encryption Standard. This underlines the importance of high nonlinearity. Also, Jakobsen and Knudsen [48] identified an attack on block ciphers with functions having small algebraic degree.

The set of symmetric Boolean functions is another small but interesting subclass of Boolean functions. In such functions the output bit depends only on the number of 1's in the input vector. These functions have received considerable attention as evident from [21, 80, 40, 118].

Next we list the properties of Boolean functions which we consider here. The formal definitions of all these cryptographic properties are discussed in Chapter 2.

- Balancedness

- Nonlinearity
- Algebraic Degree
- Correlation Immunity
- Symmetry
- Strict Avalanche Criteria
- Propagation Characteristics

We here study the combinatorial aspects of Boolean functions with these properties and construct functions with strong cryptographic significance which were not known earlier. These can be used efficiently in design of private key cryptosystems. Moreover, we study theoretical aspects of such constructions which include enumeration and identification of new sets.

1.1 Thesis Plan

In Chapter 2, we provide a brief outline of existing research and show how our work fits in that framework. Chapter 3 to Chapter 7 present the contributions of this thesis. Chapter 3 is the merged version of the communications [63, 64]. Chapters 4 to 7 are partially based on the publications [100, 66, 65, 67] respectively. We summarize the dissertation in Chapter 8. Also several open problems are presented in that Chapter.

In Chapter 3 we concentrate on a particular subset of Boolean functions called the symmetric Boolean functions. We discuss new results related to the nonlinearity of symmetric functions. A closed form expression for the Walsh transform of an arbitrary symmetric Boolean function is presented. We completely characterize the symmetric functions with maximum nonlinearity and show that the maximum nonlinearity of n-variable symmetric function can be $2^{n-1} - 2^{\lfloor \frac{n-1}{2} \rfloor}$. Moreover, new classes of symmetric balanced and symmetric correlation immune functions are considered.

It is well known that the Boolean functions to be used in cryptographic applications should have the balancedness property. In Chapter 4 we initially provide a randomized heuristic to construct balanced Boolean functions on n variables ($n \geq 15$ and odd) with nonlinearity strictly greater than $2^{n-1} - 2^{\frac{n-1}{2}}$. For such functions the algebraic degree is also maintained at its maximum, $n-1$. For odd $n \leq 13$, we construct balanced functions with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ and algebraic degree $n-1$. Moreover, we design optimized 1-resilient functions with currently best known nonlinearity. Our results are extended to construct highly nonlinear balanced and 1-resilient functions for both odd and even n. In

this chapter we also consider propagation characteristics and strict avalanche criteria. Our constructions provide balanced functions with these properties which maintain very high nonlinearity which could not be achieved earlier. We show new ways of using bent and Patterson-Wiedemann [88] functions in construction of cryptographically significant Boolean functions.

The set of correlation immune Boolean functions can be partitioned into several disjoint subsets with respect to the Hamming weights of their output column. In Chapter 5 it is shown that the number of n variable correlation immune functions of Hamming weight $2a + 2$ is strictly greater than the number of such functions of weight $2a$ for $2a < 2^{n-1}$. The proof of this seemingly intuitive result turns out to be quite involved. The technique also relates the enumeration problem of correlation immune functions to the enumeration problem of balanced correlation immune functions and provides a closed form expression for the number of correlation immune functions.

Next we concentrate on some constructions of small subsets of correlation immune functions in Chapter 6. Further, we consider the subset of correlation immune functions which satisfy one or more of a few other conditions e.g. balancedness, nondegeneracy and nonaffinity.

In Chapter 7 we provide a construction method using a small set of recursive operations for a large class of highly nonlinear, resilient Boolean functions optimizing Siegenthaler's inequality. Comparisons to previous constructions show that better nonlinearity can be obtained by our method. Our technique can be used to construct functions on large number of input variables with simple hardware implementation. We provide a special representation for such functions so that they can be implemented with low cost pipelined architecture. Moreover, the architecture is programmable and can be dynamically reconfigured to compute different functions of the class.

1.2 Prerequisites

It is assumed that the reader is familiar with undergraduate level combinatorics, linear algebra and concept of Boolean functions. The reader is referred to [68] for basic material in Boolean circuits (for Section 7.9, Chapter 7). It is not mandatory to have previous knowledge about cryptography and the properties of Boolean functions considered here.

Chapter 2

Background

In this chapter we provide a brief introduction to the existing research in related areas and identify how our work fits in that framework. The memoryless combining function to be used in the stream cipher system should be selected with care to provide proper security. Similarly if we interpret the S-boxes in the block cipher systems as a set of Boolean functions then also design of such functions need to be proper. This motivates the study on the cryptographic properties of Boolean functions. Most of the time we interpret a Boolean function as the output column of the truth table. A truth table contains tabulation of all possible combinations of input values and their corresponding outputs. The following provide an example of a 3 variable Boolean function. Note that the input variables X_3, X_2, X_1 are tabulated in each row. The function is represented in the rightmost column. For an n-variable Boolean function the truth table contains n columns for inputs, 1 column for output and 2^n rows for all the enumerations of the input variables.

X_3	X_2	X_1	f
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

By Ω_n we denote the set of all n-variable Boolean functions, that is the set of 2^{2^n} distinct binary strings of length 2^n.

Definition 2.0.1 *For binary strings S_1, S_2 of same length λ, we denote by $\#(S_1 = S_2)$ (respectively $\#(S_1 \neq S_2)$), the number of places where S_1 and S_2 are equal (respectively unequal).*

The Hamming distance between S_1, S_2 is denoted as $d(S_1, S_2)$, i.e., $d(S_1, S_2) = \#(S_1 \neq S_2)$.

The Walsh Distance is defined as, $wd(S_1, S_2) = \#(S_1 = S_2) - \#(S_1 \neq S_2)$. Note that, $wd(S_1, S_2) = \lambda - 2\,d(S_1, S_2)$.

Also the Hamming weight or simply the weight of a binary string S means the number of 1's in S. This is denoted by $wt(S)$.

Next we define balancedness.

Definition 2.0.2 *A function $f \in \Omega_n$ is said to be balanced if its output column in the truth table contains equal number of 0's and 1's (i.e. $wt(f) = 2^{n-1}$).*

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

2.1 Algebraic Degree

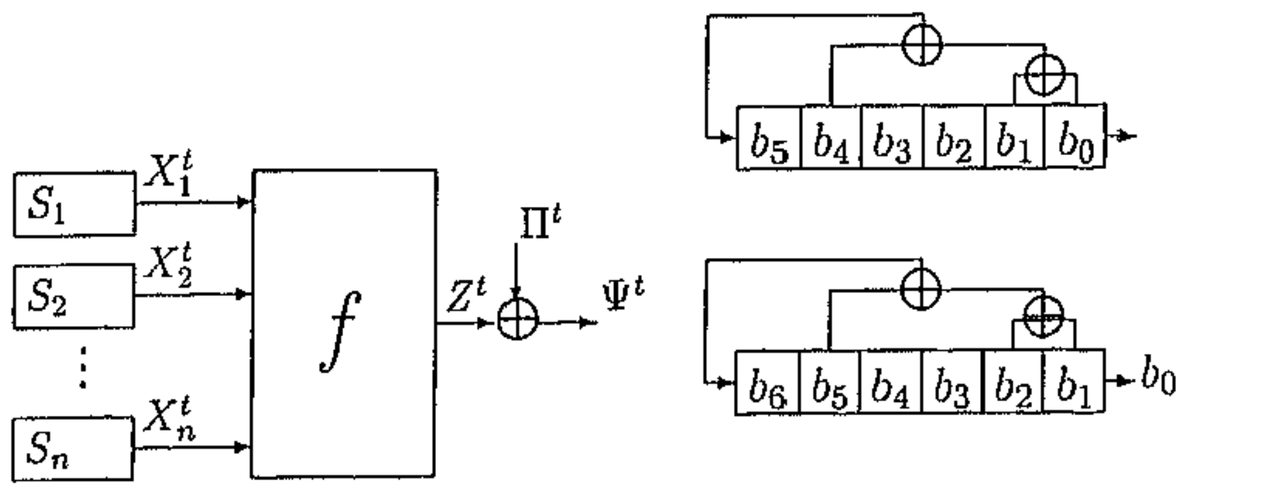
Apart from truth table, another way of representing a Boolean function is by its algebraic normal form. Note that we denote the GF(2) sum as $\oplus$.

Definition 2.1.1 *By algebraic normal form we mean the GF(2) sum of products representation of a Boolean function, which denotes the GF(2) sum of all distinct k-th order products $(0 \leq k \leq n)$ of n binary variables. Thus, a Boolean function $f(X_1, X_2, \ldots, X_n)$ can be represented as*

$$a_0 \oplus (\bigoplus_{i=1} a_i X_i) \oplus (\bigoplus_{1 \leq i \neq j \leq n} a_{ij} X_i X_j) \oplus \ldots \oplus a_{12\ldots n} X_1 X_2 \ldots X_n$$

where the coefficients $a_0, a_{ij}, \ldots, a_{12\ldots n} \in \{0, 1\}$.

The number of input variables in the highest order product term with nonzero coefficient is called the algebraic degree, or simply degree.



(a) Generalized Scheme *(b) LFSR : one step evolution*

Figure 2.1: LFSR based encryption scheme

In the stream cipher model we consider here, the combining function f is so chosen that it increases the linear complexity [96] of the resulting key stream. High algebraic degree provides high linear complexity [97, 31]. First we describe the cryptographic scheme and after that we will explain the concept of linear complexity and its relation to the algebraic degree of a Boolean function.

The cryptographic paradigm we consider here is a private key system. The sender and receiver both have the same key. The sender encrypts the message with a key and the receiver decrypts the cipher with the same key. The attacker taps the channel between the sender and receiver and grabs a portion of the cipher. The target of the attacker is then to find out the key from the cipher he possesses. See Figure 2.1a for the scheme. At the sender side, the message text is encoded to binary stream using some coding scheme. Depending on the key a binary keystream is generated. Each bit of the message is XOR-ed (addition over GF(2)) to a corresponding keystream bit and a cipherbit is generated. This cipher bit stream is transmitted through the communication channel. The receiver side possesses the same key as the sender. Hence at the receiver side same keystream bit sequence is generated. Each bit of the cipher stream is XOR-ed with the corresponding keystream bit and thus the message is recovered.

Linear Feedback Shift Registers (LFSR) are used as running key subgenerators in stream ciphers. See Figure 2.1b for one step evolution of an LFSR. Note that $b_i \in \{0, 1\}$. The LFSR in the Figure 2.1b is of length 6. It implements a recurrence relation of the form $x_n = x_{n-2} \oplus x_{n-5} \oplus x_{n-6}$ where $x_i \in \{0, 1\}$ and $\oplus$ represents the GF(2) sum (addition mod 2). The initial condition of the LFSR is $b_5b_4b_3b_2b_1b_0$. The state after one clock is $b_6b_5b_4b_3b_2b_1$ where $b_6 = b_4 \oplus b_1 \oplus b_0$ and the output is b_0. The recurrence relation may also be represented by the *connection polynomial* [76, Page 196], which is $1 + x^2 + x^5 + x^6$ for the LFSR in Figure 2.1b. The LFSRs are used as pseudorandom bit generators in cryptographic hardware.

The n subgenerators $S_i, i = 1, 2, \ldots, n$ are usually realized by LFSRs, where the feedback connection polynomials are taken to be primitive [31, 96] over GF(2). The initial conditions of the subgenerators (LFSRs) are $k_1, k_2, \ldots, k_n$ respectively, where each k_i is a bit string of length d_i. Let X be a random variable (r.v.) having the distribution given by $Prob(X = 0) = Prob(X = 1) = \frac{1}{2}$. Each subgenerator S_i produces an independent and identically distributed (i.i.d.) sequence $X_i^1, X_i^2, X_i^3, \ldots$, where $X_i^t \in \{0, 1\}\, \forall t$ and $1 \leq i \leq n$, follows the same distribution as X. A Boolean function f is used as a nonlinear combining function on n inputs $X_1, X_2, \ldots X_n$ with $Z = f(X_1, X_2, \ldots, X_n)$ (see Figure 2.1a). The keystream sequence $Z^1, Z^2, Z^3, \ldots$ is determined by $Z = f(X_1, X_2, \ldots, X_n)$. The plaintext message sequence $\Pi^1, \Pi^2, \Pi^3, \ldots$ is obtained from r.v. Π. This message sequence is bitwise XOR-ed with $Z^1, Z^2, Z^3, \ldots$ to produce the ciphertext sequence $\Psi^1, \Psi^2, \Psi^3, \ldots$ (r.v. Ψ). Deciphering is done by performing a bitwise XOR of ciphertext sequence with the keystream sequence produced by the same f and same initial conditions $k_1, k_2, \ldots, k_n$ of the LFSRs $S_1, S_2, \ldots, S_n$. The LFSRs are basically pseudorandom bit generators. The initial condition of an LFSR is the seed. Thus, the key for this private key system is the initial conditions of the n LFSRs $k_1, k_2, \ldots, k_n$.

Thus, the encryption is $\Psi = \Pi \oplus Z$ and the decryption is $\Pi = \Psi \oplus Z$. In a known plaintext attack the cryptanalyst is considered to have the plaintext and corresponding ciphertext. Thus, he has full access of some part of the running key as $Z = \Pi \oplus \Psi$. Whatever be the length of the keystream obtained by the adversary, the subsequent keystream must be unpredictable to him. Since the LFSRs are basically finite state machines, for a specific subgenerator S_i, there always exists some positive integer $q_i > 0$ such that $X_i^{q_i+j} = X_i^j$, for all $j > 0$. The value q_i is called the period of the keystream produced by the LFSR S_i. A necessary requirement for randomness of such a keystream is a long periodicity. Note that when we combine several keystreams using a Boolean function, then the resulting keystream may have better unpredictability than the individual streams. This depends on the algebraic degree of the Boolean function. In this direction we need to explain the *linear complexity*. The linear complexity of a binary keystream is the length of the shortest LFSR which is able to generate the keystream. For secure stream cipher systems large linear complexity is necessary.

Now we provide a simple example (see [96]) for more details). Let us consider a function $f(X_1, X_2, X_3, X_4) = X_1X_2 \oplus X_3X_4$. If the linear complexities of four LFSRs S_1, S_2, S_3, S_4 are 5, 6, 7, 8 respectively, then the linear complexity of the output sequence Z will be $5 \times 6 + 7 \times 8 = 86$. This clearly underlines the need for high algebraic degree of a Boolean function. Also, Jakobsen and Knudsen [48] showed that the functions used in the S-boxes need to possess high algebraic degree.

The maximum algebraic degree achievable for an n-variable Boolean function is n. However, such a function is not balanced. The maximum algebraic degree of a balanced function is $n - 1$. Moreover, if a Boolean function possesses the property correlation immunity, then

the upper bound on algebraic degree reduces further. In this dissertation we provide Boolean functions with maximum possible algebraic degree alongwith other important cryptographic criteria.

2.2 Nonlinearity

Another important cryptographic property for a Boolean function is high nonlinearity. A function with low nonlinearity is prone to *Best Affine Approximation* (BAA) [31, 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 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 [61].

Definition 2.2.1 *A Boolean function f of n variables is said to be linear if f can be expressed as $f = a_1X_1 \oplus a_2X_2 \oplus \ldots \oplus a_nX_n$, where $a_i \in \{0,1\}$. A Boolean function f of n variables is said to be affine if f can be expressed as $f = \bigoplus_{i=1}^{n} a_iX_i \oplus b$, where $b \in \{0,1\}$ for all i. We denote the set of all n-variable affine functions by $L(n)$. A Boolean function f of n variables is said to be nonlinear if f is not affine. The measure of nonlinearity of an n variable function f is $nl(f) = min_{g \in L(n)}(d(f,g))$, i.e. the distance from the set of all n-variable affine functions.*

Rothaus [95] has shown that for even n, the maximum nonlinearity achievable by a Boolean function on n variables is $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$. Such functions are called bent functions and their combinatorial properties have been studied [95, 61, 31, 15, 17]. A simple construction method for bent functions on $n = 2p$ variables from [95] is $h(X_1, \ldots, X_p, Y_1, \ldots, Y_p) = \bigoplus_{i=1}^{p} X_iY_i \oplus g(Y_1, \ldots, Y_p)$ where $g \in \Omega_p$ is arbitrary. The bent functions are known to be unbalanced. Moreover, for $n \geq 4$ the algebraic degree for a bent function is at most $\frac{n}{2}$. The question of maximum possible nonlinearity for balanced functions on even number of variables is open. For example it is not yet known whether there exists a balanced Boolean function with 8 variables having nonlinearity 118 (balanced functions with nonlinearity 116 are known).

For odd n, the class of functions with maximum nonlinearity has not been characterized (see [4, 83, 88] for important results in this area). For odd $n \leq 7$, the maximum possible nonlinearity is known and the value is $2^{n-1} - 2^{\frac{n-1}{2}}$. Mykkeltveit [83] has proved that the

maximum nonlinearity of 7-variable Boolean function is 56. Later a simpler version of the proof was provided by Hou [45]. In one of the pioneering papers in this direction, Patterson and Wiedemann [88, 89] provided construction of functions with nonlinearity strictly greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd $n \geq 15$.

Finding the maximum nonlinearity of a Boolean function on n variables is the equivalent problem of finding the covering radius of first order Reed-Muller code $R(1, m)$ [61]. This field is extremely well studied and some of the important papers in this area are [10, 9, 59, 60, 46].

We investigate the Boolean functions with maximum nonlinearity having the balancedness property. The functions provided in [89] are not balanced. Balanced functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ has been constructed for odd $n \geq 29$ [104, 32]. Construction of highly nonlinear balanced functions has also been considered in [18, 77, 78, 79, 86].

We use a randomized heuristic to construct for the first time balanced functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for $n = 15, 17, 19, 21, 23, 25, 27$. We use the functions provided in [89] as the basic input to our algorithm. Earlier these functions [89] were used to obtain balanced functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$, for odd $n \geq 29$ [104]. Construction of highly nonlinear balanced functions modifying a general class of bent functions has been considered by Dobbertin [32]. We provide an interlinked recursive algorithm for obtaining currently best known nonlinearity for balanced Boolean functions.

The algebraic degree of highly nonlinear balanced functions for $n \leq 9$ has been considered in [33, 34]. In [33], balanced functions of 7 variables with nonlinearity 56, algebraic degree 6 and balanced functions of 9 variables with nonlinearity 240, algebraic degree 7 has been reported. The method needs an exhaustive search over a subclass of Boolean functions called the idempotents and the search method is infeasible for more number of variables. Our result generalizes the results of [33]. We show that it is possible to construct balanced n (odd) variable function with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ and algebraic degree $n - 1$. Moreover, we provide construction of balanced n (odd $n \geq 15$) variable function with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ and algebraic degree $n - 1$.

We use the existing highly nonlinear functions (bent functions for even number of input variables and bent concatenation and Patterson-Wiedemann functions for odd number of variables) and modify them to obtain highly nonlinear functions possessing other important cryptographic properties.

2.3 Correlation Immunity

It is not sufficient to use functions with high algebraic degree, balancedness and high nonlinearity, since certain types of functions (even with such properties) are susceptible to other

kinds of attacks. An important *divide-and-conquer* attack on such systems was proposed by Siegenthaler [110]. Let us provide a brief description of it.

If the function f and connection polynomials of the LFSRs are known beforehand (see Figure 2.1), ciphertext only attacks need to identify the initial conditions $k_1, k_2, \ldots, k_n$ for the n LFSRs $S_1, S_2, \ldots, S_n$. Let M_i be the number of different initial conditions k_i for the LFSR S_i. So the total number of different initial conditions for the generator is $M = \prod_{i=1}^{n} M_i$. We assume that the LFSR polynomials are known. This assumption is realistic since knowledge of hardware circuit in application may be leaked which in turn identify the connection polynomial of the LFSRs. If the connection polynomial is not known, all primitive polynomials of degree d_i for each subgenerator S_i need to be checked. The size of the ith LFSR being d_i, $M_i = 2^{d_i} - 1$ (all zero condition need not be considered). This implies that it is almost impossible to check all the keys by a brute force ciphertext only attack.

Siegenthaler [110] proposed a divide and conquer attack in this kind of scenario. If the function f is not properly chosen, a cryptanalysis scheme may separately attack each subgenerator S_i individually to find the initial condition k_i. So in this case the number of trials is $M' = \sum_{i=1}^{n} M_i$. We here outline the strategy of such an attack [110]. The target is to find the initial condition of an LFSR using the ciphertext only. We assume that the connection polynomial of the LFSR is known.

Outline of the Statistical Attack:

1. N cipher bits $\Psi^1, \Psi^2, \ldots, \Psi^N$ are available.

2. Let $Prob(Z = X_i) = q_i \neq \frac{1}{2}$ for some input variable X_i corresponding to the function f. If $Prob(\Pi = 0) \neq \frac{1}{2}$, which is true in almost all practical coding schemes, then $Prob(\Psi = X_i) = \delta = Prob(Z = X_i)Prob(\Pi = 0) + Prob(Z \neq X_i)Prob(\Pi = 1) \neq \frac{1}{2}$.

3. We test the following hypothesis

$$H_0 : \theta = \frac{1}{2} \; vs \; H_1 : \theta = \delta \; (\delta \neq \frac{1}{2}).$$

where θ is the probability of success of the Bernoulli random variable Y with

$$\begin{aligned} Y^t &= 1, \quad \text{if } X_i^t = \Psi^t \\ Y^t &= 0, \quad \text{otherwise.} \end{aligned} \tag{2.1}$$

4. The actual test is carried out in the following manner. For each nonzero initial condition of S_i, we generate N bits $X_i^1, X_i^2, \ldots, X_i^N$.

   (a) Set, $count = \sum_{j=1}^{N} Y^j = \#(\Psi^j = X_i^j), 1 \leq j \leq N$.

   (b) An estimate of θ is $\frac{count}{N}$, which we use to test the hypothesis.

(c) If θ is away from $\frac{1}{2}$, the initial condition will be considered as a feasible one.

5. The feasible initial conditions constitute the set of feasible keys for S_i.

It should be noted here that what we check is the value of $Prob(\Psi = X_i)$. If it is away from $\frac{1}{2}$, then we suspect that the corresponding initial condition may be a feasible key. If $Prob(\Psi = X_i) = \frac{1}{2}$, then the r.v. Ψ and X_i are independent and the corresponding initial condition will not be considered feasible. Siegenthaler [110, Equation 3] considered the measure of correlation between Ψ and X_i as $N - 2 \times \#(\Psi^j \neq X_i^j), 1 \leq j \leq N$. Note that $N - 2 \times \#(\Psi^j \neq X_i^j) = \#(\Psi^j = X_i^j) - \#(\Psi^j \neq X_i^j)$. Thus, if we consider the expression $E = \frac{1}{N}(\#(\Psi^j = X_i^j) - \#(\Psi^j \neq X_i^j)), 1 \leq j \leq N$, then it works similar to the correlation coefficient. If $\Psi = X_i$, then $E = 1$. If $\Psi = 1 \oplus X_i$, then $E = -1$. If $\#(\Psi^j = X_i^j) = \#(\Psi^j \neq X_i^j) = \frac{N}{2}$, then $E = 0$. Basically what we need to consider is the dependence between Ψ and X_i. We need to clarify two important points at this stage.

- In step 3 the hypothesis is $H_0 : \theta = \frac{1}{2}$ *vs* $H_1 : \theta = \delta$ $(\delta \neq \frac{1}{2})$. We have $Prob(\Psi = X_i) = \delta = Prob(Z = X_i)Prob(\Pi = 0) + Prob(Z \neq X_i)Prob(\Pi = 1)$. Thus, given the value of $Prob(Z = X_i)$ and $Prob(\Pi = 0)$, it is possible to calculate δ and hence either $H_1 : \theta < \frac{1}{2}$ or $H_1 : \theta > \frac{1}{2}$ can be used. Thus, it is possible to design sharper one sided test.
- If $Prob(Z = X_i) = \frac{1}{2} \forall i$, then this method fails. The reason is $Prob(\Psi = X_i) = \frac{1}{2} \forall i$ and hence θ is not away from $\frac{1}{2}$ even if for the feasible solutions. That is the method fails due to the independence of the keystream X_i and the ciphertext Ψ.

Siegenthaler [109] himself introduced a class of Boolean functions, the set of correlation immune functions, which can resist such attacks.

Definition 2.3.1 *We consider that the set $\{X_1, X_2, \ldots, X_n\}$ of n random binary variables assumes values from $GF(2^n)$ with independent equiprobable distributions.*

A function $Z = f(X_1, X_2, \ldots, X_n)$ is mth order correlation immune [109] if the mutual information [43]

$$I(X_{i_1}, X_{i_2}, \ldots, X_{i_m}; Z) = 0$$

for all possible choices of m distinct variables $X_{i_1}, X_{i_2}, \ldots, X_{i_m} \in \{X_1, X_2, \ldots, X_n\}$, with $1 \leq m \leq n-1$.

From [43], this is equivalent to

$$Prob(Z = 1 \mid X_{i_1} = C_{i_1}, X_{i_2} = C_{i_2}, \ldots, X_{i_m} = C_{i_m}) = Prob(Z = 1)$$

for each of the combinations $C_{i_1}, C_{i_2}, \ldots, C_{i_m} \in \{0, 1\}$.

Moreover, if f is balanced then f is called m-resilient.

This Definition 2.3.1 is based on the statistical independence of Z and $X_{i_1}, X_{i_2}, \ldots, X_{i_m}$. However, the definition does not itself provide a clear idea about how such a function is resistant against divide and conquer attack. The divide and conquer attack depends on how the output of a function is related to its inputs. Thus, we provide an equivalent definition of correlation immune Boolean function which helps in formulating the modified attack. The crucial issue is the correlation immunity of a function f against such divide and conquer attacks, which naturally leads to the following definition.

Definition 2.3.2 *A Boolean function f of n variables $X_1, X_2, \ldots, X_n$, with $Z = f(X_1, X_2, \ldots X_n)$, is defined as mth order $(1 \leq m \leq n-1)$ correlation immune if*

$$Prob(Z = X_{i_m} \mid X_{i_1} = C_{i_1}, X_{i_2} = C_{i_2}, \ldots, X_{i_{m-1}} = C_{i_{m-1}}) = \frac{1}{2}$$

for all possible choices of m distinct variables $X_{i_1}, X_{i_2}, \ldots, X_{i_m} \in \{X_1, X_2, \ldots, X_n\}$ and each of the combinations $C_{i_1}, C_{i_2}, \ldots, C_{i_{m-1}} \in \{0, 1\}$.

From simple calculations of conditional probability, it is clear that Definition 2.3.1 and Definition 2.3.2 are equivalent.

If $m = 1$, $Prob(Z = X_i) = \frac{1}{2}, 1 \leq i \leq n$. A function is correlation immune if it is correlation immune of order 1 or more and it is non correlation immune if it is not correlation immune. That is, a function is non correlation immune if there exists some i, $1 \leq i \leq n$, such that $Prob(Z = X_i) \neq \frac{1}{2}$. The attack proposed by Siegenthaler [110] works on non correlation immune functions only. However, the method can be easily extended for cryptanalysis of correlation immune functions. The modified attack is as follows.

Let us consider $Prob(Z = X_i) = \frac{1}{2}$ for all i. However, for nonlinear functions, we will always get a conditional probability which is away from half. Suppose, $Prob(Z = X_{i_{m+1}} \mid X_{i_1} = C_{i_1}, X_{i_2} = C_{i_2}, \ldots, X_{i_{m-1}} = C_{i_{m-1}}, X_{i_m} = C_{i_m}) \neq \frac{1}{2}$ for some indices $i_1, i_2, \ldots, i_m, i_{m+1}$ with $1 \leq i_1 < i_2 < \ldots < i_m < i_{m+1} \leq n$, and some combination $C_{i_1}, C_{i_2}, \ldots, C_{i_{m-1}}, C_{i_m} \in \{0, 1\}$. Then it is possible to attack the subgenerator $S_{i_{m+1}}$ in a method similar to the above.

Let, $A = \{t : X^t_{i_1} = C_{i_1}, X^t_{i_2} = C_{i_2}, \ldots, X^t_{i_{m-1}} = C_{i_{m-1}}, X^t_{i_m} = C_{i_m}\}$. Here, θ is the probability of success of the Bernoulli random variable Y, for $t \in A$, with

$$\begin{aligned} Y^t &= 1, \quad \text{if } X^t_{i_{m+1}} = \Psi^t \\ Y^t &= 0, \quad \text{otherwise.} \end{aligned} \tag{2.2}$$

The test is then carried out analogous to step 4 and 5 of the previous test. In this case the number of combinations checked to find the correct initial condition for the subgenerator $S_{i_{m+1}}$ corresponding to the input $X_{i_{m+1}}$ is $M_{i_1} M_{i_2} \ldots M_{i_m} M_{i_{m+1}}$. This is less than $\prod_{i=1}^{n} M_i$ if

$m+1 < n$. Thus the immunity of a particular input $X_{i_{m+1}}$ of f may not be the maximum possible in the sense that one need not check all the $\prod_{i=1}^{n} M_i$ initial conditions.

Correlation immune and resilient functions $f : S^n \rightarrow S^r$ (where S is a set with s elements and $n \geq r \geq 1$) have extensively discussed by Gopalakrishnan in his doctoral dissertation [39]. The application of resilient Boolean functions with one or more outputs in stream cipher systems has been discussed in [39]. Moreover, it has been shown that the resiliency property is also important in design of S-boxes in block cipher systems (see [39, Page 41-42]). The study [39] contains results on orthogonal arrays and their applications in construction of binary and non-binary resilient functions. Important results in this area can also be found in [21, 36, 112, 5, 114, 41, 120]. However, in this dissertation we concentrate on single output Boolean functions only.

An important tool for the analysis of a Boolean function is its Walsh transform, which we define next.

Definition 2.3.3 *Let* $\overline{X} = (X_1, \ldots, X_n)$ *and* $\overline{\omega} = (\omega_1, \ldots, \omega_n)$ *both belong to* $\{0,1\}^n$ *and* $\overline{X}.\overline{\omega} = X_1\omega_1 \oplus \ldots \oplus X_n\omega_n$. *Let* $f(\overline{X})$ *be a Boolean function on* n *variables. Then the Walsh transform* F *of* $f(\overline{X})$ *is a real valued function over* $\{0,1\}^n$ *that can be defined as* $F(\overline{\omega}) = \sum_{\overline{X} \in \{0,1\}^n} (-1)^{f(\overline{X}) \oplus \overline{X}.\overline{\omega}}$.

Xiao and Massey [42] has provided a spectral characterization of correlation immune functions using Walsh transform. A function $f(X_1, \ldots, X_n)$ is mth order correlation immune iff its Walsh transform F satisfies $F(\overline{\omega}) = 0$, for $1 \leq wt(\overline{\omega}) \leq m$. Note that balanced mth order correlation immune functions are called m-resilient functions and if f is balanced then $F(\overline{0}) = 0$. Thus, a function $f(X_1, \ldots, X_n)$ is m-resilient iff its Walsh transform F satisfies $F(\overline{\omega}) = 0$, for $0 \leq wt(\overline{\omega}) \leq m$.

2.3.1 Construction

First we concentrate on the construction of balanced Boolean functions with high order of correlation immunity, high algebraic degree and high nonlinearity. Siegenthaler in [109] proved a fundamental inequality relating the number of variables n, order of correlation immunity m and algebraic degree d of a Boolean function: $m + d \leq n$. Moreover, if the function is balanced then $m + d \leq n - 1$. Since it is natural to use balanced functions in stream cipher systems we concentrate only on resilient functions. A resilient Boolean function is said to be optimized with respect to Siegenthaler's inequality (or in short optimized) if $m + d = n - 1$. We provide construction methods for optimized functions having high nonlinearities. The functions are built using a small set of recursive operations and hence functions on large number of variables are easy to implement using nominal hardware.

Construction procedures for correlation immune (CI) functions were first described by Siegenthaler in [109]. The methods described in [109] are recursive, where a function of $(n+1)$ variables is built from two functions of n variables. Siegenthaler considered two different kinds of constructions, one where the order of correlation immunity remains constant and the other where the order of correlation immunity increases by one at each step.

Further attempts at construction was made by Camion et al in [12], where construction procedure for a certain subset of correlation immune functions were described. Seberry et al [105], also provided a method of constructing the same subset as in [12] of correlation immune functions. Importantly, they also considered the algebraic degree, nonlinearity and propagation characteristics of their construction method [105]. The functions constructed in [105] has good nonlinearity for non optimized functions. However, for optimized functions the lower bound on nonlinearity of [105] decreases. We interpret the *direct construction* method proposed in [105] in a simpler manner as a concatenation of linear functions. This interpretation simplifies the proofs related to correlation immunity and nonlinearity.

Evolutionary techniques are applied in [78] to design first order correlation immune balanced functions with high nonlinearity. The technique considers the output column of the function as a string and applies genetic algorithm to manipulate this string. Therefore this technique is difficult to apply to construct functions on n variables for even moderate values of n. Moreover, it is not clear whether these functions optimize the Siegenthaler's inequality. More importantly, without relaxing the optimization criterion of the Siegenthaler's inequality, we can achieve better nonlinearity than in [78]. Favourable results than [78] can also be found using construction procedure in [105].

In another approach to the problem, Filiol and Fontaine [33, Section 5] describe a method to construct functions which achieve a good trade-off between nonlinearity, balancedness, degree and correlation immunity. They identify a 7 variable function f with nonlinearity 56 and degree 6. Using f, in [33, Section 5], they construct a balanced 9 variable function g with nonlinearity 224, correlation immunity of order 2 and degree 6, where they use a technique which was first introduced in [109, Section VI], and later in [12, Corollary 4.1]. The function g is optimized with respect to Siegenthaler's inequality. The key of this construction is the existence of f which was obtained by exhaustive search over a particular subset (the idempotents) of Boolean functions. However, it seems infeasible to carry out such an exhaustive search for functions with larger number of input variables. Construction methods of resilient functions have also been considered in [19, 86].

We use concatenation techniques and introduce generic construction functions which recursively build a correlation immune function of $(n+1)$ variables from two correlation immune functions of n variables. We initially start with bent functions which are modified a little to get optimized algebraic degree. A sequence of such constructors is applied to build correlation immune functions of desired orders from non correlation immune balanced

Boolean functions with high nonlinearity. The degree of the resulting function is same as that of the initial function. The method can easily be extended to design functions with moderate to large number of input variables using a special representation of the constructed Boolean functions. The actual trade-off between nonlinearity and correlation immunity of this construction is explicit. Moreover, we use linear algebraic techniques to provide construction of n (odd) variable optimized 1-resilient functions (i.e. algebraic degree $n-2$) with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$. When these functions are used as the initial functions in our recursive technique, we get improved nonlinearity than our basic method.

Current stream cipher systems use a nonlinear Boolean function to combine the output of a small number (8 to 10) of LFSRs to produce a keystream which is used to encrypt the data. With the present speed of computers, using Boolean functions of a small number of input variables (and hence a small number of LFSRs) is sufficient. However, considering the extreme pace in which the present day hardware is advancing, within a few years the known attacks [110, 73, 31, 75, 20, 22, 90, 37, 52, 51, 49, 50, 13, 84, 62] may perform significantly well against the stream cipher systems which use functions on a small number of input variables. More importantly, the recent attacks [52, 51] have shown how the weight restriction on the polynomials can be lifted using coding theory based techniques. Thus, it is important to implement cryptographically strong functions on large number of input variables at a nominal hardware cost. We use a pipelined architecture for this.

We exploit the pipelining technique for increasing speed which also provide a low cost cascadable architecture. Using the special representation a Boolean function f of n ($= m+k+1$) variables is built from a Boolean function h of k variables. The value of k is low enough so that h can be easily implemented (either as a lookup table or by a combinational circuit). Given the value of $h(X_k, \ldots, X_1)$ we present two algorithms to compute the value of $f(X_n, \ldots, X_{k+1}, X_k, \ldots, X_1)$ in $(m+1)$ steps. A direct hardware implementation of any one of the algorithms would require $(m+1)$ clocks to compute $f(X_n, \ldots, X_1)$. However, using a simple and efficient *store and forward* pipelined architecture having $(m+1)$ stages, the value of $f(X_n, \ldots, X_1)$ is available at each clock after an initial latency period of $(m+1)$ clocks. Another significant advantage of our implementation is that the hardware for one function can be easily reconfigured dynamically to compute another function of this large class.

2.3.2 Weight Distribution & Some Small Subsets

Enumeration of combinatorial objects has always been an interesting exercise. In particular, the set of Boolean functions is one of the extremely rich combinatorial structures. The history of research on Boolean functions begins as early as in 1938 when Murnaghan [82] noted that the number of n variable Boolean functions is 2^{2^n}. In [111], Slepian pointed out that operations of permuting and/or complementing one or more of the n input variables

of a Boolean function constitute a finite group and he considered the enumeration problem of such distinct groups. More recently, Denev and Tonchev [27] provided an asymptotic estimate for the number of equivalence classes under transformation of the form $f(\overline{X}) \rightarrow f(A\overline{X} + B) + l(\overline{X})$, $\overline{X} = (X_1, \ldots, X_n)$, where f is a Boolean function on n variables, A an $n \times n$ nonsingular matrix, B is a vector and l a linear function. The interest of such enumeration is still alive as evident from a very recent paper [91], where Pippenger used information theoretic techniques in presenting enumeration results on monotone Boolean functions.

Enumeration problems for cryptographically significant Boolean functions including correlation immunity have got a lot of attention as evident from [80, 107, 28, 118, 81]. The currently best known estimate of the cardinality of correlation immune Boolean functions is provided by Denisov [28].

We provide results on weight distribution of correlation immune functions. It is easy to see that for $a < 2^{n-1}$, the number of n-variable functions of weight a is less than the number of n-variable functions of weight $a + 1$. This follows from simple properties of binomial coefficients. It is then intuitive to expect same kind of results for correlation immune functions as well. However, this seemingly intuitive result turns out to be quite difficult to be proved. We show that the number of n variable CI functions of Hamming weight $2a+2$ is strictly greater than the number of such functions of weight $2a$ for $2a < 2^{n-1}$. The combinatorial structure of CI functions revealed here relates the enumeration problem of CI functions to the enumeration problem of balanced CI functions. Moreover, we provide interesting results on the CI functions relating the Hamming distance between the function (interpreted as a binary string of length 2^n) and its reverse binary string. Our results provide a different direction in comparison with the existing results for enumeration of correlation immune functions.

From the results of weight distribution of CI functions it is clear that though closed form expressions for the number of correlation immune functions can be achieved, calculating those expressions are extremely time consuming and getting exact enumeration is a nontrivial task. An asymptotic estimate of the cardinality of correlation immune functions has been provided in [28]. Currently known results regarding the lower and upper bounds on correlation immune functions appear in [80, 118, 81]. Let A_n be the set of CI Boolean functions. Then the lower bound is provided by the recursive relation $\mid A_n \mid \geq \mid A_{n-1} \mid^2$ and the upper bound is given by $\mid A_n \mid \leq \sum_{j=0}^{2^{n-2}} \binom{2^{n-2}}{j}^4$. In [102, 26], exhaustive search techniques for all the correlation immune(CI) functions have been proposed. However, use of these methods are practically infeasible for $n \geq 7$. At this point it needs to be mentioned that the best known enumeration result is the one proposed by Denisov [28], though it is not constructive. The result is as follows [28], $A_n \sim \dfrac{2^{2^n}}{2\exp\{(\ln\sqrt{\frac{\pi}{2}} + (\frac{n}{2} - 1)\ln 2)n\}}$.

We also identify some interesting subclasses of correlation immune functions and estimate the cardinality of those subclasses. These results are not as good as the result by Denisov [28] from enumeration point of view, but they provide some in depth understanding about constructive methods of correlation immune functions. Our motivation here is to find out some interesting subclasses of correlation immune functions and their interrelationship. These techniques are consecutively used in Chapter 7 for constructing cryptographically significant resilient Boolean functions.

2.4 Symmetric Functions

A particularly interesting subclass of Boolean functions is the set of symmetric functions, where the output of the function depends only on the number of ones in the input variables. Cryptographic and combinatorial properties of symmetric functions have been studied in [11, 21, 80, 31, 40, 118].

Definition 2.4.1 *A function $f(X_1, \ldots, X_n)$ is symmetric if the output of f is same for all the vectors $\{X_1, \ldots, X_n\}$ of same Hamming weight.*

One important advantage of symmetric functions is that an n-variable function can be represented using only $(n+1)$-bits, storing the values corresponding to the inputs of different weights $0, 1, \ldots, n$. We denote the reduced form of a symmetric Boolean function f by $re(f)$, a binary string of length $(n+1)$, defined as follows. For $0 \leq i \leq n$, $re(f)[i] = f(X_1, \ldots, X_n)$, where $wt(X_1, \ldots, X_n) = i$.

First we discuss the results related to symmetric Boolean functions. In [31], elementary symmetric Boolean functions were defined and the expression of Walsh transform for such elementary symmetric functions and also for strict majority logic functions were derived. However, closed form expression for the Walsh transform of arbitrary symmetric Boolean functions is not known, which we present here. For a general Boolean function on n variables, the best known algorithm to compute the Walsh transform requires $O(n2^n)$ time [2]. From the closed form expression of the Walsh transform we show that for symmetric functions this can be done in $O(n^3)$ time.

Next we study the symmetric functions with maximum nonlinearity. Ding, Xiao and Shan [31] have described a construction of (elementary) symmetric bent functions. We characterize the class of symmetric bent functions by showing that all the possible symmetric bent functions can be represented by contiguous odd length binary substring of (1100)*. That is given any symmetric bent function f, $re(f)$ is a contiguous odd length binary substrings of (1100)*. This class includes the constructions of [31].

Maximum nonlinearity of Boolean functions on odd number of variables is an important theoretical question and this is related to the covering radius of the Reed-Muller code [61, 23, 46]. In [89] it has been shown that there exists functions on n ($n \geq 15$, odd) variables with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$. We show that this does not hold for the set of symmetric Boolean functions and the maximum nonlinearity that can be achieved in the class is $2^{n-1} - 2^{\frac{n-1}{2}}$. In fact, for odd n the characterization of maximum nonlinear functions is similar to that of the symmetric bent functions. These can also be represented by contiguous $(n+1)$ length binary substrings of $(1100)^*$. For $n \geq 2$, we show that the algebraic degree of symmetric functions with maximum nonlinearity is 2 irrespective of the number of input variables.

The study of balanced correlation immune symmetric functions was initiated by Chor, Goldreich, Hastad, Friedman, Rudich and Smolensky [21] and later work on this set was carried out by Gopalakrishnan, Hoffman and Stinson [40]. However, these works considered the properties of balancedness and correlation immunity together. Here we separately consider symmetric balanced and symmetric correlation immune functions. The analysis of these sets leads to interesting equations on binomial coefficients. We provide sufficient conditions for the solutions of these equations and hence obtain new constructions of such functions. The enumerative implications of these constructions improve upon the lower bounds of these sets provided earlier by Mitchell [80] and Yang and Guo [118].

2.5 Propagation Characteristics & Strict Avalanche Criteria

The security of DES type of block ciphers are analysed by viewing the S-boxes as a set of Boolean functions. In addition to balancedness, nonlinearity and high algebraic degree, *propagation characteristics (PC) and strict avalanche criteria(SAC)* are important properties of Boolean functions to be used in S-boxes.

Definition 2.5.1 *Let $\overline{X}$ be an n tuple $X_1, \ldots X_n$ and $\overline{\alpha} \in \{0,1\}^n$. A function $f \in \Omega_n$ is said to satisfy*

1. *SAC if $f(\overline{X}) \oplus f(\overline{X} \oplus \overline{\alpha})$ is balanced for any $\overline{\alpha}$ such that $wt(\overline{\alpha}) = 1$.*
2. *$SAC(k)$ if any function obtained from f by keeping any k input bits constant satisfies SAC.*
3. *$PC(l)$ if $f(\overline{X}) \oplus f(\overline{X} \oplus \overline{\alpha})$ is balanced for any $\overline{\alpha}$ such that $1 \leq wt(\overline{\alpha}) \leq l$.*

4. *PC(l) of order k if any function obtained from f by keeping any k input bits constant satisfies PC(l).*

Preneel et al [93, 92] provided basic construction techniques for Boolean functions with these properties. In [93], it has been shown that for balanced SAC(k) functions on n variables, $deg(f) \leq n - k - 1$. Recently in [58], balanced SAC(k) functions on n variables with $deg(f) = n - k - 1$ has been identified for $n - k - 1 =$ odd. However, construction of such functions for $n - k - 1 =$ even has been left as an open problem. In [58] balanced SAC(k) functions with high algebraic degree and in [103] highly nonlinear balanced SAC functions have been proposed. However, balanced SAC(k) functions with both high algebraic degree and high nonlinearity have not been studied. PC(l) of order k functions with good nonlinearity and algebraic degree have been reported in [58]. These properties have also been considered in [35, 105, 104, 24, 106, 57, 119, 44].

We use a linear transformation on the input variables of the functions provided in [89] to obtain PC(1) functions on n variables (for odd $n \geq 15$) with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^{\frac{n-15}{2}}$. This establishes the existence of PC(1) functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd $n \geq 15$.

We improve the algebraic degree and nonlinearity results of the PC(l) of order k functions reported in [58]. Motivated by the construction methods of SAC(k) functions in [58], we introduce a new cryptographic criterion called the *restricted balancedness* of Boolean functions and show that certain types of bent functions satisfy this property. Also we modify the functions provided by Patterson and Wiedemann [88, 89] to obtain restricted balancedness while keeping the nonlinearity unchanged. This requires an in-depth study of the internal structure of the functions provided in [89].

For the first time we consider the properties of balancedness, SAC(k), algebraic degree and nonlinearity together. We construct balanced (using the functions with restricted balancedness) SAC(k) functions in Ω_n with maximum possible algebraic degree $n - k - 1$ and very high nonlinearity for $k \leq \frac{n}{2} - 1$. This also shows that there exists balanced SAC(k) functions on n variables with $deg(f) = n-k-1 =$ even, which was posed as an open question in [58]. We also present an interesting result on 1-resilient functions satisfying PC(k).

Chapter 3

Symmetric Boolean Functions

In this chapter first we discuss the nonlinearity issues of symmetric Boolean functions. We obtain a closed form expression for the Walsh Transform of an n-variable Symmetric Boolean function and characterize the set of symmetric Boolean functions (on both odd and even number of input variables) having maximum possible nonlinearity. We show that for an even integer n, any n-variable symmetric bent function can be represented by a contiguous $(n+1)$ length substring of $(1100)^*$. We also show that for an odd integer n, any n-variable symmetric Boolean function can achieve maximum nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ and can be represented similarly by a contiguous $(n+1)$ length substring of $(1100)^*$. This completely settles the problem of maximum nonlinearity for the set of symmetric Boolean functions.

Next new subsets of symmetric balanced and symmetric correlation immune functions are identified. The method involves interesting identities on binomial coefficients and highlights the combinatorial richness of these classes. As a consequence of our constructive techniques, we improve upon the existing lower bounds on the cardinality of these sets.

3.1 Introduction

A particularly interesting subclass of Boolean functions is the set of symmetric functions, where the output of the function depends only on the weight of the input vector. Combinatorial properties of symmetric functions have been studied in [21, 31, 40]. Here we obtain a closed form expression for the Walsh transform of a symmetric function and characterize the set of symmetric functions achieving the maximum possible nonlinearity.

In [31], elementary symmetric Boolean functions were defined as

$$f_{(k)}(X_1, \ldots, X_n) = \bigoplus_{1 \le i_1 < \ldots < i_k \le n} X_{i_1} X_{i_2} \ldots X_{i_k},$$

which denotes the $GF(2)$ sum of all distinct k-th order products of n binary variables. The expression of Walsh transform for such elementary symmetric functions and also for strict majority logic functions were derived in [31]. However, closed form expression for the Walsh transform of arbitrary symmetric Boolean functions is not known, which we provide here. For a general Boolean function on n variables, the best known algorithm to compute the Walsh transform requires $O(n2^n)$ time [2]. From the closed form expression of the Walsh transform we show that for symmetric functions this can be done in polynomial time. The Walsh transform for the $\overline{0}$ vector can be computed in $O(n)$ time. The Walsh transform for the nonzero vectors of n distinct weights can be viewed as multiplication of an $n \times (n+1)$ matrix D_n by an $(n+1)$ vector. The construction of the matrix D_n takes $O(n^3)$ time.

One of the most important classes of Boolean functions is the set of bent functions [95, 17]. Bent functions achieve the maximum nonlinearity and exist only if the number of input variables is even. For even n, the maximum nonlinearity of an n-variable (bent) function is $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$. Rothaus [95] has shown that the maximum algebraic degree of n-variable (for even $n \geq 4$) bent functions can be at most $\frac{n}{2}$. Ding, Xiao and Shan [31] describe the following construction of (elementary) symmetric bent functions: $f_{(2)}(X_1, \ldots, X_n) = \bigoplus_{1 \leq i \neq j \leq n} X_i X_j$.

Here we characterize the class of symmetric bent functions by showing that all the possible symmetric bent functions can be represented by contiguous odd length binary substrings of $(1100)^*$. This class includes the constructions of [31]. In a recent survey by Carlet [16], enumeration of bent functions and identifying their algebraic normal forms were posed as important questions. Our work provides complete answer to these questions for the class of symmetric Boolean functions.

Maximum nonlinearity of Boolean functions on odd number of variables is an important theoretical question and this is related to the covering radius of the Reed-Muller code [61, 23, 46]. In [89] it has been shown that there exists functions on n ($n \geq 15$, odd) variables with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$. We show that this does not hold for the set of symmetric Boolean functions and the maximum nonlinearity that can be achieved in this class is $2^{n-1} - 2^{\frac{n-1}{2}}$. In fact, for odd n the characterization of maximum nonlinear functions is similar to that of the symmetric bent functions. These can also be represented by contiguous $(n+1)$ length binary substrings of $(1100)^*$. For $n \geq 2$, we show that the algebraic degree of symmetric functions with maximum nonlinearity is 2 irrespective of the number of input variables.

The study of balanced correlation immune symmetric functions was initiated by Chor, Goldreich, Hastad, Friedman, Rudich and Smolensky [21] and later work on this set was carried out by Gopalakrishnan, Hoffman and Stinson [40]. However, these works considered the properties of balancedness and correlation immunity together. Here we separately consider symmetric balanced and symmetric correlation immune functions. The analysis of these sets

leads to interesting equations on binomial coefficients. We provide sufficient conditions for the solutions of these equations and hence obtain new constructions of such functions. The enumerative implications of these constructions improve upon the lower bounds of these sets provided earlier by Mitchell [80] and Yang and Guo [118]. We need the following definition and notation provided by Mitchell [80] for comparing our results with [80, 118].

Definition 3.1.1 *Let $f(X_1, \ldots, X_n)$ be a Boolean function.*
C1. Balancedness. *The function f is balanced if the number of ones in its output column is equal to the number of zeros.*
C2. Nonaffinity. *The function f is linear/affine if it can be written as $f(X_1, \ldots, X_n) = \bigoplus_{i=1}^{i=n} a_i X_i \oplus b$, where $a_i, b \in \{0, 1\}$. The function f is nonaffine if it is not linear/affine.*
C3. Nondegeneracy. *The function f is degenerate if there exists at least one variable $X_i \in \{X_1, \ldots, X_n\}$, such that,*
$f(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots, X_n) = f(X_1, \ldots, X_{i-1}, 1, X_{i+1}, \ldots, X_n)$. The function f is non-degenerate if it is not degenerate.
C4. Correlation Immunity. *The function f is correlation immune (CI) if $Prob(f = X_i) = \frac{1}{2}$ (i.e. $\#(f = X_i) = 2^{n-1}$), $\forall i$, $1 \leq i \leq n$.*
C5. Symmetry. *The function f is symmetric if $f(X_1, \ldots, X_n)$ is same for all the vectors $\{X_1, \ldots, X_n\}$ of same Hamming weight.*

By Ω_n we denote the set of all n-variable Boolean functions. Also $A_n(i_1, \ldots, i_t)$ is the set of all n-variable Boolean functions having the properties $Ci_1, \ldots, Ci_t$. Note that $A_n(5)$ denotes the set of all symmetric Boolean functions on n variables.

Chor et al [21] had raised the question of whether $A_n(1, 2, 4, 5)$ is empty. This question was settled by Gopalakrishnan, Hoffman and Stinson in [40], where they provided constructions of functions in the set $A_n(1, 2, 4, 5)$. Since the only degenerate symmetric functions are the identity 0 and 1 functions, this also shows that $A_n(1, 2, 3, 4, 5) \neq \emptyset$, which solves the open question posed by Mitchell in [80].

Here we provide construction of new functions in the sets $A_n(1, 2, 3, 5)$ and $A_n(2, 3, 4, 5)$. These can be used to provide improved lower bounds on the sizes of the corresponding sets. Our constructions characterize the "sporadic" examples in $A_n(1, 2, 3, 5)$ reported by Brüer [11] and Mitchell [80]. We clearly identify the richness of the sets $A_n(1, 2, 3, 5)$ and $A_n(2, 3, 4, 5)$ which were not explored earlier.

Before proceeding further we introduce a few notations. A function $f \in \Omega_n$ can be represented by a binary string of length 2^n, the output column of its truth table. By $f = f_1 f_2$, we mean the n-variable function f constructed by concatenating the output columns of f_1 and f_2, which are $(n-1)$ variable functions. The reverse of a binary string s is denoted by s^r and the bitwise complement of s is denoted by s^c. It is also clear that an n-variable symmetric

Boolean function can be represented by an $(n+1)$ length binary string where the i-th location contains the output corresponding to the input vector of Hamming weight i. For a binary string s of length k, $s[i]$ denotes the i-th bit for $0 \leq i \leq k-1$. Let $f \in A_n(5)$ be represented by a binary string of length 2^n. We denote the reduced form of f by $re(f)$, a binary string of length $(n+1)$, defined as follows. For $0 \leq i \leq n$, $re(f)[i] = f(X_1, \ldots, X_n)$, where $wt(X_1, \ldots, X_n) = i$. Similarly, given a binary string g of length $(n+1)$, we define extension of g, $ex(g)$, to be a symmetric function f of length 2^n as $f(X_1, \ldots, X_n) = g[wt(X_1, \ldots, X_n)]$. The maps $re(f)$ and $ex(g)$ are one-to-one correspondences between n-variable symmetric Boolean functions and binary strings of length $(n+1)$.

3.2 Walsh Transform

Walsh transform provides a frequency domain analysis of Boolean functions. In [31], expressions for Walsh transform of elementary symmetric Boolean functions and strict majority logic functions were obtained. However, we here provide an expression for Walsh transform of arbitrary symmetric Boolean function. We use the $(n+1)$ length bit string $re(f)$ to obtain the formula.

First let us recapitulate that given two binary strings S_1 and S_2 of same length λ, the Walsh distance between S_1, S_2 is defined as $wd(S_1, S_2) = \#(S_1 = S_2) - \#(S_1 \neq S_2)$. Note that, $wd(S_1, S_2) = \lambda - 2d(S_1, S_2)$, where $d(S_1, S_2)$ is the Hamming distance between S_1 and S_2. The following result provides the relationship between Walsh distance and Walsh Transform.

Proposition 3.2.1 *$F(\overline{\omega}) = wd(f, \bigoplus_{i=1}^{i=n} \omega_i X_i)$, where F is the Walsh transform of f.*

Proof :

$$\begin{aligned} F(\overline{\omega}) &= \sum_{\overline{X} \in \{0,1\}^n} (-1)^{f(\overline{X}) \oplus \overline{X}.\overline{\omega}} \\ &= \#\{f(\overline{X}) = \overline{X}.\overline{\omega}\} - \#\{f(\overline{X}) \neq \overline{X}.\overline{\omega}\} \\ &= \#\{f(\overline{X}) = \bigoplus_{i=1}^{i=n} \omega_i X_i\} - \#\{f(\overline{X}) \neq \bigoplus_{i=1}^{i=n} \omega_i X_i\} \\ &= wd(f, \bigoplus_{i=1}^{i=n} \omega_i X_i). \end{aligned}$$

■

Thus finding the Walsh transform values is equivalent to finding the Walsh distance between the function f and all linear functions. Given an n-bit vector $\overline{\alpha} = (\alpha_1, \ldots, \alpha_n)$ we define the linear function $l_{\overline{\alpha}}(X_1, \ldots, X_n)$, represented by $\overline{\alpha}$ as follows.

$$l_{\overline{\alpha}}(X_1, \ldots, X_n) = \alpha_1 X_1 \oplus \ldots \oplus \alpha_n X_n.$$

Any n-variable linear function can be written as $l_{\overline{\alpha}}$ for some n-bit vector $\overline{\alpha}$. By the weight of an n-variable linear function we mean the weight of the corresponding $\overline{\alpha}$.

Proposition 3.2.2 *Let $f(X_1, \ldots, X_n) \in A_n(5)$. Let l_1 and l_2 be two n-variable linear functions of the same weight. Then $wd(f, l_1) = wd(f, l_2)$. Consequently, $F(\overline{\omega}) = F(\overline{\omega}')$ if $wt(\overline{\omega}) = wt(\overline{\omega}')$, where F is the Walsh transform of f.*

This shows that it is sufficient to compute Walsh transform for $(n+1)$ distinct $\overline{\omega}$, having weights 0 to n. We now show how this can be done. Let us consider the truth table of a Boolean function. In the truth table the column corresponding to X_j is at the left hand side of X_i if $j > i$.

Proposition 3.2.3 *Let $l_k = X_n \oplus \ldots \oplus X_{n-k+1} \in \Omega_n$. Then $\#\{l_k = 1 \mid wt(\overline{X}) = i\} = \sum_{j \text{ odd}}^{k} \binom{k}{j}\binom{n-k}{i-j}$ and $\#\{l_k = 0 \mid wt(\overline{X}) = i\} = \sum_{j \text{ even}}^{k} \binom{k}{j}\binom{n-k}{i-j}$.*

Proof : Note that, $\#\{l_k = 1 \mid wt(\overline{X}) = i\} = \#\{\overline{X} = X_n \ldots X_{n-k+1}X_{n-k} \ldots X_1 \mid wt(X_n \ldots X_{n-k+1}) = j \text{ odd}, wt(X_n \ldots X_1) = i\}$ and $\#\{l_k = 0 \mid wt(\overline{X}) = i\} = \#\{\overline{X} = X_n \ldots X_{n-k+1}X_{n-k} \ldots X_1 \mid wt(X_n \ldots X_{n-k+1}) = j \text{ even}, wt(X_n \ldots X_1) = i\}$. In the truth table of an n variable function, there are $\binom{n}{i}$ input rows, where i variables out of the n variables have the value 1 and the other $(n-i)$ variables have the value 0. Choose j variables from k variables $\{X_n, \ldots, X_{n-k+1}\}$ and $i-j$ variables from $n-k$ variables $\{X_{n-k}, \ldots, X_1\}$. If j is odd, l_k is 1 and if j is even, l_k is 0. ■

Using this we get the following formula for Walsh transform of a symmetric function. It is interesting to note that the expression for the Walsh transform resembles closely the expression for Krawtchouk polynomial [55].

Theorem 3.2.1 *Let $f \in A_n(5)$ and $WTS(f) = \{i \mid re(f)[i] = 1\}$. For $1 \leq k = wt(\overline{\omega}) \leq n$, $F(\overline{\omega}) = 2 \sum_{i \in WTS(f)} \sum_{j=0}^{k} (-1)^{j+1} \binom{k}{j}\binom{n-k}{i-j}$ and $F(\overline{0}) = 2^n - 2 \sum_{i \in WTS(f)} \binom{n}{i}$.*

Proof : We have, $d(\overline{0}, f) = wt(f) = \sum_{i \in \{WTS(f)\}} \binom{n}{i}$. Thus, $F(\overline{0}) = 2^n - 2 \sum_{i \in WTS(f)} \binom{n}{i}$.

Now we prove the result for the case where $1 \leq wt(\overline{\omega}) \leq n$. Note that $F(\overline{\omega}) = wd(f, \bigoplus_{i=1}^{i=n} \omega_i X_i)$. Thus we get, $F(\overline{\omega}) =$

$$= \sum_{i \in WTS(f)} [\sum_{j \text{ odd}} \binom{k}{j}\binom{n-k}{i-j} - \sum_{j \text{ even}} \binom{k}{j}\binom{n-k}{i-j}] + \sum_{i \notin WTS(f)} [\sum_{j \text{ even}} \binom{k}{j}\binom{n-k}{i-j} - \sum_{j \text{ odd}} \binom{k}{j}\binom{n-k}{i-j}]$$

$$= \sum_{i \in WTS(f)} [\sum_{j\ odd} \binom{k}{j}\binom{n-k}{i-j} - \sum_{j\ even} \binom{k}{j}\binom{n-k}{i-j}] - \sum_{i \notin WTS(f)} [\sum_{j\ odd} \binom{k}{j}\binom{n-k}{i-j} - \sum_{j\ even} \binom{k}{j}\binom{n-k}{i-j}].$$

Also, we have, $wd(\overline{1}, l) = 0$ for any linear function $l \in \Omega_n$. Thus,

$= \sum_{0 \le i \le 2^n} [\sum_{j\ odd} \binom{k}{j}\binom{n-k}{i-j} - \sum_{j\ even} \binom{k}{j}\binom{n-k}{i-j}] = 0$. Hence,

$$\sum_{i \in WTS(f)} [\sum_{j\ odd} \binom{k}{j}\binom{n-k}{i-j} - \sum_{j\ even} \binom{k}{j}\binom{n-k}{i-j}] = - \sum_{i \notin WTS(f)} [\sum_{j\ odd} \binom{k}{j}\binom{n-k}{i-j} - \sum_{j\ even} \binom{k}{j}\binom{n-k}{i-j}].$$

So we get,

$$F(\overline{\omega}) = 2 \sum_{i \in WTS(f)} [\sum_{j\ odd} \binom{k}{j}\binom{n-k}{i-j} - \sum_{j\ even} \binom{k}{j}\binom{n-k}{i-j}] = 2 \sum_{i \in WTS(f)} \sum_{j=0}^{k} (-1)^{j+1} \binom{k}{j}\binom{n-k}{i-j}. \quad \blacksquare$$

Given a function $f \in \Omega_n$, calculating $F(\overline{\omega})$ for a specific $\overline{\omega} \in \{0,1\}^n$ takes $O(2^n)$ time. Fast Walsh Transform takes $O(n2^n)$ time for calculating $F(\overline{\omega})$ for all $\overline{\omega} \in \{0,1\}^n$.

Our analysis shows that for $f \in A_n(5)$, we do not need to calculate the Walsh transform for all 2^n possible $\overline{\omega} \in \{0,1\}^n$. It is sufficient to calculate the Walsh transform for only $(n+1)$ distinct $\overline{\omega}$ with different Hamming weights $0, 1, 2, \ldots, n-1, n$. Given f, calculating $F(\overline{0}) = 2^n - 2 \sum_{i \in WTS(f)} \binom{n}{i}$ takes $O(n)$ time. However, the algorithm for calculating Walsh transform for the n nonzero $\overline{\omega}$ of distinct weights can be seen as the multiplication of an $n \times (n+1)$ matrix D_n by an $(n+1) \times 1$ matrix, which is the $(n+1)$ length bit vector $re(f)$, written as a column. The (k,i)-th location of the matrix D_n contains $\sum_{j=0}^{k} (-1)^{j+1} \binom{k}{j}\binom{n-k}{i-j}$, for $1 \le k \le n$ and $0 \le i \le n$. Construction of the matrix D_n takes $O(n^3)$ preprocessing time and then calculation of Walsh transform for the n nonzero $\overline{\omega}$ of distinct weights takes $O(n^2)$ time. Thus, once the matrix D_n is ready, the algorithm needs $O(n^2)$ time.

3.3 Bent Functions

In this section we provide complete characterization of symmetric bent functions of n variables for even n. We prove that $f \in A_n(5)$ is bent iff $re(f)$ is a contiguous $(n+1)$ length substring of $(1100)^*$. Before proceeding we require the following well known property of bent functions [95].

Theorem 3.3.1 *A function $f \in \Omega_n$ is bent for even n iff all its Walsh transform values are $\pm 2^{\frac{n}{2}}$.*

If f is bent, then f^c, f^r and $f^{rc} = (f^r)^c = (f^c)^r$ are all bent. Let $f \in A_n(5)$. We have, $re(f^r) = (re(f))^r$, $re(f^c) = (re(f))^c$ and $re(f^{rc}) = (re(f))^{rc}$. If $f \in A_n(5)$ is bent and

$g = re(f)$, then $ex(g^r), ex(g^c)$ and $ex(g^{rc})$ are also symmetric bent functions. Also we will require the following result. The result uses the concept of *dual* [95, 16] of a bent function.

Definition 3.3.1 *For even n, let $f \in \Omega_n$, be a bent function. Let $\hat{f} \in \Omega_n$ be such that $2^{-\frac{n}{2}} F(\overline{\omega}) = (-1)^{\hat{f}(\overline{\omega})}$, where F is the Walsh transform of f. Then $\hat{f}$ is called the dual of f.*

Proposition 3.3.1 *For even n, let $f_1, f_2 \in \Omega_n$ be two bent functions such that $wd(f_1, l) = -wd(f_2, l)$ for each $l \in L(n)$. Then $f_1 = f_2^c$.*

Proof : The proof follows from the following two facts.
(i) The dual of a dual of a bent function is the function itself.
(ii) The dual of a bent function is also a bent function. ■

The following result plays a crucial role in characterizing symmetric bent functions.

Lemma 3.3.1 *Let m be even and $F \in \Omega_{m+2}$ be bent such that $F = f_0 f_1 f_2 f_3$, where, $f_0, f_1, f_2, f_3 \in \Omega_m$ and $f_1 = f_2$. Then (1) f_0, f_1, f_2, f_3 are bent and (2) $f_0 = f_3^c$.*

Proof : For F, Consider that the input variables are $\{X_1, \ldots, X_{m+2}\}$ and for f_0, f_1, f_2, f_3, the input variables are $\{X_1, \ldots, X_m\}$. In the truth table, the column corresponding to X_{i+1} is at the left hand side of the column corresponding to X_i.

Now consider l, a linear function of k variables, $0 \leq k \leq m$, and the variables are selected from $\{X_1, \ldots, X_m\}$. Since F is bent, all of its Walsh transform values are $\pm 2^{\frac{m+2}{2}}$. From Proposition 3.2.1, Theorem 3.3.1 and considering the other two variables X_{m+1}, X_{m+2} we have
$wd(F, llll) = \pm x$, $wd(F, ll^c ll^c) = \pm x$, $wd(F, lll^c l^c) = \pm x$ and $wd(F, ll^c l^c l) = \pm x$, where $x = 2^{\frac{m+2}{2}}$. Let, $wd(f_0, l) = a_l$, $wd(f_1, l) = wd(f_2, l) = b_l$ and $wd(f_3, l) = c_l$. Hence we get the following three equations.

$$\begin{array}{lllll} a_l & +2b_l & +c_l = & \pm x & (1) \\ a_l & & -c_l = & \pm x & (2) \\ a_l & -2b_l & +c_l = & \pm x & (3) \end{array}$$

Here we have to select the $+$ or $-$ sign for x. Thus, we have total 8 options. Let us consider the solutions taking all the combinations of $+$ and $-$ for x in the three equations.

The possible values of b_l are $\pm\frac{x}{2}, 0$. The value of b_l is 0 iff the sign of x in equation(1) and equation(3) are same. We claim that b_l cannot be 0. In contradiction if $b_l = 0$, for some l, then $\sum_{\overline{\omega} \in \{0,1\}^m} (wd(f_1, \bigoplus_{i=1}^{m} \omega_i X_i))^2 < 2^{2m}$. However, this is not possible, as from Parseval's Theorem [31, Page 15] we get, for any function $f \in \Omega_m$, $\sum_{\overline{\omega} \in \{0,1\}^m} (wd(f, \bigoplus_{i=1}^{m} \omega_i X_i))^2 = 2^{2m}$. Hence, for each l, the sign of x in equation(1) and equation(3) must be opposite.

For opposite signs of x in equation(1) and equation(3), the possible values of a_l, b_l, c_l are $\pm\frac{x}{2} = \pm 2^{\frac{m}{2}}$ for every l. Thus, from Theorem 3.3.1, $f_0, f_1, f_2, f_3 \in A_m(5)$ are all bent. Also, it is important to note that the signs of a_l and c_l are opposite for every l. Thus, from Proposition 3.3.1, we get that $f_0 = f_3^c$. ■

Proposition 3.3.2 *Let $F \in A_{m+2}(5)$ and $F = f_0 f_1 f_2 f_3$, where $f_0, f_1, f_2, f_3 \in \Omega_m$. Then (1) $f_0, f_1, f_2, f_3 \in A_m(5)$, (2) $f_1 = f_2$, (3) $re(f_0)$ is the initial $(m+1)$ bits of $re(F)$, whereas $re(f_3)$ is the last $(m+1)$ bits. The middle $(m+1)$ bits of $re(F)$, leaving the initial and final bits constitute $re(f_1) = re(f_2)$.*

From Lemma 3.3.1 and Proposition 3.3.2, we get the following result regarding symmetric bent functions.

Lemma 3.3.2 *Let m be even and $F \in A_{m+2}(5)$ be bent. Let $F = f_0 f_1 f_2 f_3$, where, $f_0, f_1, f_2, f_3 \in \Omega_m$. Then (1) f_0, f_1, f_2, f_3 are also symmetric bent, (2) $f_0 = f_3^c$, (3) $f_1 = f_2$.*

In [1], it has been observed that if f_i, f_j, f_k, f_h are bent functions then $f_i f_j f_k f_h$ is a bent function, provided one of f_i, f_j, f_k, f_h is the complement of another, and the other two are same, e.g. $f_h = f_k^c$, and $f_i = f_j$. They use it to provide a recursive lower bound on the number of bent functions.

What we prove in Lemma 3.3.2 is the other direction. We only take F as a bent function and consider it as a concatenation (sequence) of f_0, f_1, f_2, f_3. Here we do not even consider that f_0, f_1, f_2, f_3 are bent. We only use the condition $f_1 = f_2$. Then we prove the properties of f_0, f_1, f_2, f_3.

We are now ready to present the characterization of symmetric bent functions.

Theorem 3.3.2 *For even n, $F \in A_n(5)$ is bent iff $re(F)$ is a contiguous $(n+1)$ length substring of* $(1100)^*$. *Consequently, there are only four bent functions in $A_n(5)$.*

Proof : First suppose F is bent and let $re(F) = s_0 s_1 \ldots s_{n-1} s_n$, where $s_i \in \{0,1\}$, $0 \leq i \leq n$. Consider F to be a concatenation of f_0, f_1, f_2, f_3 in $A_{n-2}(5)$, i.e. $F = f_0 f_1 f_2 f_3$. Using Proposition 3.3.2, we get, $g_0 = re(f_0) = s_0 s_1 \ldots s_{n-3} s_{n-2}$, $g_1 = re(f_1) = re(f_2) = s_1 s_2 \ldots s_{n-2} s_{n-1}$ and $g_3 = re(f_3) = s_2 s_3 \ldots s_{n-1} s_n$.

Using Lemma 3.3.1, we get $f_0 = f_3^c$, which implies $g_0 = g_3^c$. Hence, $s_{i+2} = s_i^c$ for $0 \leq i \leq n-2$. Thus, if F is symmetric bent then $re(F)$ is a contiguous $(n+1)$ length substring of $(1100)^*$.

We prove the other direction by induction on the number of variables. Assume the result is true for all m-variable functions and let F be an $m+2$-variable function such that $g = re(F)$ is a $m+3$ length contiguous substring of $(1100)^*$. Then we can write

$g = s_0 s_1 \ldots s_{m-1} s_m s_{m+1} s_{m+2}$, where each $s_i \in \{0,1\}$. Since g is a contiguous substring of $(1100)^*$ it can be verified that $s_{m+1} = s_{m-1}^c$ and $s_{m+2} = s_m^c$. We define $g_0 = s_0 s_1 \ldots s_{m-1} s_m$, $g_1 = g_2 = s_1 \ldots s_{m-1} s_m s_{m+1}$ and $g_3 = s_2 \ldots s_{m+1} s_{m+2}$. Again since g is a contiguous $m+3$ length substring of $(1100)^*$, g_0, g_1, g_2, g_3 are $m+1$ length substrings of $(1100)^*$ and also $g_0 = g_3^c$. Define $f_i = ex(g_i)$ for $0 \leq i \leq 3$. Then F is concatenation of f_0, f_1, f_2 and f_3. By induction hypothesis f_i's are all bent. Now the linear functions l_1 of $m+2$ variables are of the forms $llll, ll^c ll^c, lll^c l^c, ll^c l^c l$, where l is a linear function of m variables. It now follows that $wd(F, l_1)$ is either $2wd(f_0, l)$ or $\pm 2wd(f_1, l)$. This shows that F has a two valued spectra and hence is bent.

There are exactly four distinct contiguous substrings of $(1100)^*$ of length $m+1$ and hence there are exactly 4 distinct n-variable symmetric bent function. ■

From Theorem 3.3.2 we get a complete characterization of the class of symmetric bent functions. Next we discuss a few results related to the algebraic normal form of any symmetric Boolean function and then use these to identify the algebraic normal form of symmetric bent functions.

It is well known [31] that in a symmetric Boolean function either all the k-th order terms are present or all are absent at the same time. Thus, the algebraic normal form of a symmetric Boolean function f can also be represented by an $n+1$ length bit vector $ra(f)$ (the reduced algebraic normal form of f), where $ra(f)[k] \in \{0,1\}$ and $ra(f)[k] = 0 (resp.\ 1)$ means that all the k-th order terms are absent (resp. present). For $f \in A_n(5)$, the following result relates the vectors $re(f)$ and $ra(f)$.

Theorem 3.3.3 *For $f \in A_n(5)$, let us consider, $g = re(f)$ and $q = ra(f)$. Then, $g[i] = (\sum_{k=0}^{i} q[k]\binom{i}{k}) \bmod 2$, where $0 \leq i \leq n$ and $0 \leq k \leq i$.*

Proof : Since all vectors of the same weight have the same output value it is sufficient to consider an arbitrary input vector of weight i, for $0 \leq i \leq n$. We now compute the output value corresponding to such a vector. All terms in the ANF having terms of length greater than i must necessarily evaluate to 0. Now consider terms of length k with $0 \leq k \leq i$, and $q[k] = 1$. Then exactly $\binom{i}{k}$ number of k length terms (out of the total $\binom{n}{k}$ number of k length terms in the ANF) will evaluate to 1. From this the proof follows. ■

This expression provides an algorithm to generate either g from q or q from g. If q is known, we get g from direct calculation. However, if g is known, then q need to be generated recursively. That is for calculating $q[k]$, all the values of $q[0], \ldots, q[k-1]$ need to be calculated. As example, if g is known, then $g[0] = q[0]$. For the next step, $g[1] = \sum_{k=0}^{1} q[k]\binom{1}{k} \bmod 2 = q[0] + q[1] \bmod 2$ and since $g[1], q[0]$ are known, $q[1]$ can be calculated.

In this manner all the bits of q can be calculated. Now we provide the algebraic normal form of the symmetric bent functions. In [95], it has been shown that for $n \geq 4$, the maximum possible algebraic degree for the bent functions is $\frac{n}{2}$. We show that the algebraic degree of symmetric bent functions is 2 irrespective of the number of input variables.

Theorem 3.3.4 *Let $F \in A_n(5)$ be bent, for even n. Then F can be represented as* $F(X_1, \ldots, X_n) = (\bigoplus_{1 \leq i < j \leq n} X_i X_j) \oplus b(\bigoplus_{i=1}^{n} X_i) \oplus c$, *where, $b, c \in \{0, 1\}$.*

Proof : We first consider the bent function F with $re(F) = 0011\ldots = g$. Let $q = ra(F)$. Using Theorem 3.3.3, we get $q[0] = q[1] = q[3] = 0$ and $q[2] = 1$. We now show that for $j \geq 1$, $q[4j] = q[4j+1] = q[4j+2] = q[4j+3] = 0$. We have $g[4i] = g[4i+1] = 0$ and $g[4i+2] = g[4i+3] = 1$ for $i \geq 1$. Using Theorem 3.3.3 we have, $g[4j] = 0 = (\sum_{k=0}^{4j} q[k]\binom{4j}{k}) \bmod 2 = (\binom{4j}{2} q[2] + q[4j]) \bmod 2$. Since $\binom{4j}{2} \bmod 2 = 0$ and $g[4j] = 0$, we have $q[4j] = 0$. Similarly it can be shown that $q[4j+1] = 0$. The proofs that $q[4j+2], q[4j+3]$ are zero are similar and we only show $q[4j+2] = 0$. Using Theorem 3.3.3 we have, $g[4j+2] = 0 = (\sum_{k=0}^{4j+2} q[k]\binom{4j+2}{k}) \bmod 2 = (\binom{4j+2}{2} q[2] + q[4j+2]) \bmod 2$. Since $\binom{4j+2}{2} \bmod 2 = 1$ and $q[2] = g[4j+2] = 1$, we have $q[4j+2] = 0$. Thus, we get that $q[i] = 0$ for $0 \leq i \leq n, i \neq 2$ and $q[2] = 1$. Thus F is of the form $(\oplus_{1 \leq i < j \leq n} X_i X_j)$.

Given F and g, the other three n-variable bent functions are $F^r = ex(g^r), F^c = ex(g^c), F^{rc} = ex(g^{rc})$. So it is sufficient to show that F^r has the form given in the theorem. To see this it is enough to observe that $F^r(X_1, \ldots, X_n) = F(X_1 \oplus 1, \ldots, X_n \oplus 1)$. ∎

3.4 Maximum nonlinearity for odd n

One standard way to achieve highly nonlinear functions on odd number variables is to concatenate two bent functions. The nonlinearity obtained by this process is $2^{n-1} - 2^{\frac{n-1}{2}}$. However, the concatenation of two symmetric functions need not be symmetric, so this method will in general not work for symmetric functions. Here we extend the construction of symmetric bent functions and show how to obtain highly nonlinear symmetric functions for odd number of input variables.

First we present the following result on nonlinearity.

Proposition 3.4.1 *Let $F_1, F_2 \in \Omega_{n-1}$ and $F = F_1 F_2$. If $nl(F) = 2^{n-1} - x$, $0 < x < 2^{n-2}$, then both $nl(F_1), nl(F_2) \geq 2^{n-2} - x$. Moreover, if $F_1 = F_2$, then $nl(F_1) = nl(F_2) = 2^{n-2} - \frac{x}{2}$.*

Proof : We prove, $2^{n-2}-x \leq d(F_1,l) \leq 2^{n-2}+x$ for any $l \in L(n-1)$. Since $nl(F) = 2^{n-1}-x$, we have $2^{n-1} - x \leq d(F_1F_2, ll) \leq 2^{n-1} + x$. (1)

Let $d(F_1,l) = 2^{n-2} - x - t < 2^{n-2} - x$, $t > 0$. Then from (1) we get $2^{n-2} + t \leq d(F_2,l) \leq 2^{n-2} + 2x + t$. Now, $d(F_1, l^c) = 2^{n-2} + x + t$. So, $2^{n-1} + x + 2t \leq d(F_1F_2, l^cl) \leq 2^{n-1} + 3x + 2t$, which is a contradiction to $nl(F) = 2^{n-1} - x$. Thus, $d(F_1,l) \geq 2^{n-2} - x$.

Let $d(F_1,l) = 2^{n-2}+x+t > 2^{n-2}+x$, $t > 0$. Then from (1) we get $2^{n-2}-2x-t \leq d(F_2,l) \leq 2^{n-2} - t$. Now, $d(F_1,l^c) = 2^{n-2} - x - t$. So, $2^{n-1} - 3x - 2t \leq d(F_1F_2, l^cl) \leq 2^{n-1} - x - 2t$, which is a contradiction to $nl(F) = 2^{n-1} - x$. Thus, $d(F_1,l) \leq 2^{n-2} + x$.

Hence we get $nl(F_1) \geq 2^{n-2} - x$. To see the last statement, note that if $F_1 = F_2$, then $nl(F) = 2nl(F_1)$. ■

Corollary 3.4.1 *Let $f_0, f_1, f_2, f_3 \in \Omega_{n-2}$ and $F = f_0f_1f_2f_3$. If $nl(F) = 2^{n-1} - x$, $0 < x < 2^{n-2}$, then for the function $f_jf_k \in \Omega_{n-1}$, $nl(f_jf_k) \geq 2^{n-2} - x$, $0 \leq j \neq k \leq 3$.*

Proof : First note that any function in $L(n)$, can be written as $llll$, lll^cl^c, ll^cll^c or ll^cl^cl where $l \in L(n-2)$. The proof goes in the similar way as the Proposition 3.4.1. We consider only the case for ll^cl^cl, the others being similar.

Let $d(f_1f_2, l^cl^c) = 2^{n-2} - x - t < 2^{n-2} - x$, $t > 0$. Then we get $2^{n-2} + t \leq d(f_0f_3, ll) \leq 2^{n-2} + 2x + t$. Now, $d(f_1f_2, ll) = 2^{n-2} + x + t$. So, $2^{n-1} + x + 2t \leq d(f_0f_1f_2f_3, ll^cl^cl) \leq 2^{n-1} + 3x + 2t$, which is a contradiction to $nl(F) = 2^{n-1} - x$. Thus, $d(f_1f_2, l) \geq 2^{n-2} - x$. Similarly it can be shown that $d(f_1f_2, l) \leq 2^{n-2} + x$. Thus $nl(f_1f_2) \geq 2^{n-2} - x$. ■

Now we consider the following subset of Boolean functions.

Definition 3.4.1 *Consider a function $F \in \Omega_n$ and $F = f_0f_1f_2f_3$, where $f_i \in \Omega_{n-2}$ $0 \leq i \leq 3$ and there exists at least two integers j,k, $0 \leq j \neq k \leq 3$ such that $f_j = f_k$. We call F a matched function. The set of matched functions of n variables is denoted by Δ_n.*

Note that Δ_n is a superset of the set of symmetric Boolean functions.

Proposition 3.4.2 $| \Delta_n | = 6.2^{\frac{3}{4}2^n} - 11.2^{2^{n-1}} + 6.2^{2^{n-2}}$.

Proof : Let $F = f_0f_1f_2f_3$, where $F \in \Omega_n$ and $f_0, f_1, f_2, f_3 \in \Omega_{n-2}$. We subtract the functions where f_0, f_1, f_2, f_3 are all distinct from the set of all functions of n variables. The result is then equal to $2^{2^n} - {}^{2^{2^{n-2}}}\mathcal{P}_4$, where ${}^n\mathcal{P}_r$ denotes number of permutations of n objects taken r at a time. ■

The following result relates the nonlinearity attainable for Δ_n to that attainable for Δ_{n-2}.

Lemma 3.4.1 *Let $F \in \Delta_n$, where $nl(F) > 2^{n-1} - 2^{\frac{n-1}{2}}$. Then there exists $f \in \Omega_{n-2}$ with $nl(f) > 2^{n-3} - 2^{\frac{n-3}{2}}$.*

Proof : Let $F = f_0 f_1 f_2 f_3$, where $f_i \in \Omega_{n-2}$ $0 \leq i \leq 3$. Since $F \in \Delta_n$, without loss of generality we can consider $f_1 = f_2$. Then from Corollary 3.4.1 we get $nl(f_1 f_2) > (2^{n-1} - 2^{\frac{n-1}{2}}) - 2^{n-2} = 2^{n-2} - 2^{\frac{n-1}{2}}$. Since $f_1 = f_2 = f$, say, we get $2nl(f) > 2^{n-2} - 2^{\frac{n-1}{2}}$. Since $2nl(f)$ is even, $2nl(f) \geq 2^{n-2} - 2^{\frac{n-1}{2}} + 2$ and hence $nl(f) \geq 2^{n-3} - 2^{\frac{n-3}{2}} + 1$. ■

We will use Lemma 3.4.1 negatively in the following corollary which will in turn be used in the proof of Theorem 3.4.2. The contrapositive of the above result is going to be more useful for our application and hence we state it as the following corollary.

Corollary 3.4.2 *If there does not exist any $f \in \Omega_{n-2}$ with $nl(f) > 2^{n-3} - 2^{\frac{n-3}{2}}$, then there does not exist any $F \in \Delta_n$, where $nl(F) > 2^{n-1} - 2^{\frac{n-1}{2}}$.*

Let $F \in \Delta_9$. For any $f \in \Omega_7$, it is known that $nl(f) \leq 56$ [83]. Let $n = 9$. Thus, $nl(f)$ can not be greater than $2^{n-3} - 2^{\frac{n-3}{2}}$. Then there does not exist any $F \in \Delta_n$, where $nl(F) > 2^{n-1} - 2^{\frac{n-1}{2}}$. Thus, $nl(F) \leq 240$. Note that this result can not be extended for the class Δ_n for odd $n > 9$ as the maximum nonlinearity of 9 variable Boolean functions is still not known. However, this provides an important upper bound on the nonlinearity of symmetric Boolean functions with odd number of input variables.

The following upper bound on the nonlinearity of symmetric functions (on odd number of variables) is an immediate application of Corollary 3.4.2.

Theorem 3.4.1 *Let $F \in A_n(5)$ for n odd and $n \geq 3$. Then $nl(F) \leq 2^{n-1} - 2^{\frac{n-1}{2}}$.*

Proof : We find that for the base case $n = 3$ the assertion is true. Then the induction follows from Corollary 3.4.2 as all n-variable symmetric functions are of the form $F = f_0 f f f_3$ and hence in Δ_n. ■

For odd number of variables, the characterization of functions achieving maximum nonlinearity is described in the following result.

Theorem 3.4.2 *Let $F \in A_n(5)$ for n odd. Then $nl(F) = 2^{n-1} - 2^{\frac{n-1}{2}}$ iff $re(F)$ is a contiguous $n+1$ length substring of $(1100)^*$.*

Proof : It is not difficult to see that if $re(F)$ is a contiguous $(n+1)$ length substring of $(1100)^*$, then F is a concatenation of two bent functions and hence achieves the stated nonlinearity.

Conversely, suppose $nl(F) = 2^{n-1} - 2^{\frac{n-1}{2}}$. Since F is symmetric, we can write F as $F = f_0 f f f_3$, where f_0, f, f_3 are symmetric functions of $n-2$ variables. We show by induction on odd n, that F is a contiguous $n+1$ length substring of $(1100)^*$ and also the Walsh Transform of F is three valued, $0, \pm 2^{\frac{n+1}{2}}$.

We first prove $nl(f) = 2^{n-3} - 2^{\frac{n-3}{2}}$. Since $nl(F) = 2^{n-1} - 2^{\frac{n-1}{2}}$, using Corollary 3.4.1, we have $nl(ff) \geq 2^{n-2} - 2^{\frac{n-1}{2}}$. However, $nl(ff) = 2nl(f)$ and so $nl(f) \geq 2^{n-3} - 2^{\frac{n-3}{2}}$. Also using Theorem 3.4.1 the maximum possible nonlinearity for a $(n-2)$ variable symmetric function is $2^{n-3} - 2^{\frac{n-3}{2}}$ and so $nl(f) = 2^{n-3} - 2^{\frac{n-3}{2}}$.

By induction hypothesis we can assume that $re(f)$ is a contiguous $n-1$ length substring of $(1100)^*$ and the Walsh Transform values of f are $0, \pm 2^{\frac{n-1}{2}}$. Thus the possible forms of $re(f)$ are
1) $g_1 = 001100....$
2) $g_2 = 110011....$
3) $g_3 = 01100....$
4) $g_4 = 10011....$
Let $G = re(F) = xgy$, for some $x, y \in \{0,1\}$. Also g must be one of g_1, g_2, g_3, g_4. We now show that the following must hold.
A) If $g = g_1$, then $x = 1$ and $y = b$.
B) If $g = g_2$, then $x = 0$ and $y = 1 - b$.
C) If $g = g_3$, then $x = 0$ and $y = b$.
D) If $g = g_4$, then $x = 1$ and $y = 1 - b$.
where $b = \frac{(n-1) \bmod 4}{2}$.

Note that it is sufficient to show (A) and (C). This is because $g_2 = g_1^c$ and $g_4 = g_3^c$ and $ex(xhy)$ and $ex(x^c h^c y^c)$ have the same nonlinearity for any $n-1$ length bit string h. Here we prove only (A), the proof of (C) being similar. We have to prove that the other combinations of x and y result in lower nonlinearities. If x and y have the values given in the conditions then G is an $(n+1)$ length contiguous substring of $(1100)^*$ and hence achieve the required nonlinearity.

Now we turn to the proof of (A). We only prove for the condition $n-1 \equiv 0 \bmod 4$, the case $n-1 \equiv 2 \bmod 4$ being similar. Since $n-1 \equiv 0 \bmod 4$, we have $re(F) = x00110011\ldots0011y$. Let $s_0 = re(f_0)$, $s_3 = re(f_3)$ and $t = re(f)$. Therefore
$t = 00110011\ldots0011$,
$s_0 = x00110011\ldots11001$ and
$s_3 = 011001100\ldots0011y$.
Let $s = 100110011\ldots11001$ and l be a linear function such that $wd(ex(s), l) = a$. We now rule out the three possible options except the case $x = 1, y = 0$.
Case 1 : $x = y = 0$.
Let $\#(ex(s) = l) = a_1$ and $\#(ex(s) \neq l) = a_2$. Then $a = a_1 - a_2$. Now $\#(ex(s_0) = l) = a_1 + 1$ and $\#(ex(s_0) \neq l) = a_2 - 1$ and so $wd(f_1, l) = wd(ex(s_0), l) = a + 2$. Also $wd(f_2, l) = wd(ex(s_3), l) = -a$, since $s_3 = s^c$ when $y = 0$. By induction hypothesis the Walsh Transform values of f are $0, \pm 2^{\frac{n-1}{2}}$. Let l be such that $wd(f, l) = 2^{\frac{n-1}{2}}$. Then $wd(F, llll) = wd(f_1, l) + 2wd(f, l) + wd(f_2, l) = 2 + 2^{\frac{n-1}{2}}$. Hence $d(F, llll) = 2^{n-1} - 2^{\frac{n-1}{2}} - 1 <$

$2^{n-1} - 2^{\frac{n-1}{2}}$, which contradicts $nl(F) = 2^{n-1} - 2^{\frac{n-1}{2}}$.
Case 2 : $x = 0, y = 1$.
In this case $s_3 = (s_0)^{rc}$. Let l be a nondegenerate linear function on odd number of variables and hence $l^r = l^c$. Then $wd(f_2, l) = wd(f_1^{rc}, l) = wd(f_1, l^{rc}) = wd(f_1, l) = b$ (say). Since there are exactly 2^{n-3} linear functions such that $l^r = l^c$, it cannot be the case that $wd(f, l) = 0$ for all such functions as otherwise this would violate Parseval's theorem. So we can choose l such that $wd(f, l) = \pm 2^{\frac{n-1}{2}}$. Now two cases arise
(i) $wd(f, l) = 2^{\frac{n-1}{2}}$. If $b > 0$, then consider $wd(F, llll) = 2^{\frac{n+1}{2}} + 2b$ and if $b < 0$, then consider $wd(F, l^c lll^c) = 2^{\frac{n+1}{2}} + 2b$.
(ii) $wd(f, l) = -2^{\frac{n-1}{2}}$. If $b > 0$, then consider $wd(F, l^c lll^c) = -2^{\frac{n+1}{2}} - 2b$ and if $b < 0$, then consider $wd(F, llll) = -2^{\frac{n+1}{2}} - 2b$.
Therefore either $d(F, llll) < 2^{n-1} - 2^{\frac{n-1}{2}}$ or $d(F, l^c lll^c) < 2^{n-1} - 2^{\frac{n-1}{2}}$ and so $nl(F) < 2^{n-1} - 2^{\frac{n-1}{2}}$, which is a contradiction.
Case 3 : $x = y = 1$.
In this case $wd(f_1, l) = a$ and $wd(f_1^c, l) = -a$. Let l be such that the last bit of l is 0, i.e. nondegenerate on even number of variables and hence $l^r = l$. Then $wd(f_2, l) = -a - 2$. Now combining the techniques of the above two cases we can show that $nl(F) < 2^{n-1} - 2^{\frac{n-1}{2}}$, which is a contradiction. ∎

From Theorem 3.4.2 we get a complete characterization of the class of symmetric functions with maximum nonlinearity for odd number of input variables. Similar to Theorem 3.3.4 we obtain the algebraic normal forms for such functions. The algebraic degree of symmetric functions with odd number of input variables and the algebraic degree is also 2 irrespective of the number of input variables. It should be noted that these are partially bent functions. In the following theorem we show that these functions are quadratic and it is known that any quadratic function is partially bent [14].

Theorem 3.4.3 *Let $F \in A_n(5)$ for odd n with maximum nonlinearity. Then F can be represented as $F(X_1, \ldots, X_n) = (\bigoplus_{1 \le i < j \le n} X_i X_j) \oplus b(\bigoplus_{i=1}^{n} X_i) \oplus c$, where, $b, c \in \{0, 1\}$.*

An important property of Boolean functions is its propagation characteristics defined as follows. A n-variable Boolean function f is said to satisfy $PC(k)$ if $f(\overline{X}) \oplus f(\overline{X} \oplus \overline{\alpha})$ is balanced for all $1 \le wt(\overline{\alpha}) \le k$.

Theorem 3.4.4 *For n odd, there exists balanced $F \in A_n(5)$ with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ satisfying $PC(n-1)$.*

Proof : For $n = 4m+1$ consider the $4m+2$ length string $g = 0(1100)^m 1$ and let $F = ex(g)$. Then F is of the form ff^{rc} where f is a symmetric bent function on $4m$ variables. Thus F

is balanced. Similarly, for $n = 4m + 3$ consider the $4m + 4$ length string $g = 00(1100)^m11$ and let $F = ex(g)$. Then also F is of the form ff^{rc} where f is a symmetric bent function on $4m+2$ variables. Thus F is balanced. The nonlinearity is equal to the nonlinearity achieved by the concatenation of two bent functions.

The function f is a symmetric bent function. It is well known [61] that bent functions of $(n-1)$ variables satisfy $PC(n-1)$. Since F is of the form ff^{rc}, it satisfies propagation characteristics with respect to all the n-bit vectors except the all one vector. ■

3.5 Balancedness

In [11, 80], the problem of enumerating $A_n(1,5)$ is discussed, and a lower bound on the number of balanced symmetric functions is obtained. A simple way to obtain balanced symmetric functions is provided in [80]. From the properties of binomial coefficients we know, $\sum_{i \text{ even}} \binom{n}{i} = \sum_{i \text{ odd}} \binom{n}{i} = 2^{n-1}$. This immediately gives rise to the two nondegenerate affine functions in $A_n(5)$, which are clearly balanced. If n is odd, one can form $\frac{n+1}{2}$ pairs of binomial coefficients of the form $\{\binom{n}{i}, \binom{n}{n-i}\}$, where two elements of the pair have the same value. Then choosing one element from each pair will give rise to a function in $A_n(1,5)$. This way of partitioning immediately gives rise to the following bound on $\mid A_n(1,5) \mid$.

$$\mid A_n(1,5) \mid \geq \begin{cases} 2^{\frac{n+1}{2}} & \text{if } n \text{ is odd,} \\ 2 & \text{if } n \text{ is even.} \end{cases}$$

The inequality is strict when some nontrivial partitioning is found. Brüer [11] tabulates $\mid A_n(1,5) \mid$ for odd $n \leq 17$ and obtains $\mid A_n(1,5) \mid = 2^{\frac{n+1}{2}}$ except $\mid A_{13}(1,5) \mid = 144$. Mitchell [80] has also shown that $\mid A_8(1,5) \mid > 2$ and termed these as "sporadic" examples. We here show that these are not "sporadic" and there exist infinitely many integer values of n for which we get the strict inequality. We start with the following simple result.

Proposition 3.5.1 *A symmetric function f is palindromic iff $re(f)$ is palindromic, i.e., iff $re(f)[i] = re(f)[n-i]$ i.e., iff for all input vectors of weight i and $n-i$, $0 \leq i \leq n$, f outputs the same value.*

The balanced functions constructed here are all non constant functions and hence are nondegenerate (as all nonconstant symmetric functions are nondegenerate). Thus $A_n(1,5) = A_n(1,3,5)$. Apart from the two affine functions, all the other functions in $A_n(1,3,5)$ are nonaffine. Hence $\mid A_n(1,2,3,5) \mid = \mid A_n(1,3,5) \mid - 2$. The constructions of balanced symmetric functions in the following two results provide two new subsets of symmetric balanced functions.

Theorem 3.5.1 *Let $n \equiv 2 \bmod 6$. Then it is possible to construct $f \in A_n(1,2,3,5)$. Consequently, $\mid A_n(1,5) \mid > 2$.*

Proof : If $n \equiv 2 \bmod 6$, there exists r such that $n+1 = 3r$. Note that the condition $n+1 = 3r$ holds iff $\binom{n}{r-1} + \binom{n}{n-r+1} = \binom{n}{r}$.

Consider the representation of two nondegenerate affine functions l_1 and $l_2 = 1 \oplus l_1$ in $A_n(5)$, when n is even. The $(n+1)$ bit representation of these two functions are $re(l_1) = 010101\ldots01010$ and $re(l_2) = 101010\ldots10101$. Given the value of n, if r is odd, choose $l = l_1$, else choose $l = l_2$. That is we choose the nondegenerate symmetric affine function $l \in A_n(5)$ where the output is 1 corresponding to any input of Hamming weight r. Note that the linear functions are balanced.

Since n is even it follows that $r - 1 \equiv n - r + 1 \not\equiv r \bmod 2$. Thus, in $re(l)$, r-th location contains 1 and the $(r-1)$-th and $(n-r+1)$-th locations contain 0. Now we construct a nonlinear balanced symmetric function f. Note that all the balanced symmetric functions are nondegenerate.

The bit string $re(f)$ is the same as $re(l)$ except three places, $r-1, r, n-r+1$. Thus, in $re(f)$, r-th location contains 0 and the $(r-1)$-th and $(n-r+1)$-th locations contain 1. That is we replace the weight r by weights $r-1$ and $n-r+1$. This replacement will not change the balancedness condition as $\binom{n}{r-1} + \binom{n}{n-r+1} = \binom{n}{r}$ and will give rise to nonaffine balanced functions f, f^c in $A_n(5)$. ∎

Theorem 3.5.1 justifies why Mitchell [80] got nontrivial bipartitioning for $n = 8$. This will follow for $n = 14, 20, \ldots$ and so on.

Theorem 3.5.2 *Let $n \geq 13$ be odd and $(n+3)$ a perfect square. Then $\mid A_n(1,2,3,5) \mid \geq (2^{\frac{n+1}{2}} - 2) + 2^{\frac{n+1}{2}-3}$.*

Proof : The term $(2^{\frac{n+1}{2}} - 2)$ comes from trivial partitioning except the two affine functions. Next we explain the second term.

Here we have to consider the condition $\binom{n}{r-1} + \binom{n}{r+2} = \binom{n}{r} + \binom{n}{r+1}$. It can be checked using simplification that $\binom{n}{r-1} + \binom{n}{r+2} = \binom{n}{r} + \binom{n}{r+1}$ iff $(n-2r-1)^2 = n+3$. Note that for $n \geq 13$, $(n-2r-1)^2 = n+3$ implies $r+2 < \frac{n-1}{2}$.

Let $f \in A_n(5)$. Now consider the pairs
$\{\binom{n}{r-1}, \binom{n}{n-r+1}\}, \{\binom{n}{r}, \binom{n}{n-r}\}, \{\binom{n}{r+1}, \binom{n}{n-r-1}\}, \{\binom{n}{r+2}, \binom{n}{n-r-2}\}$. Apart from these four pairs we have $\frac{n+1}{2} - 4$ pairs, where we choose one term from each pair providing $2^{\frac{n+1}{2}-4}$ options. Note that an assignment $\{re(f)[r-1] = 1, re(f)[n-r+1] = 0\}$, $\{re(f)[r+2] = 1, re(f)[n-r-2] = 0\}$,
$\{re(f)[r] = 0, re(f)[n-r] = 1\}$, $\{re(f)[r+1] = 0, re(f)[n-r-1] = 1\}$ will provide a function

which is balanced and has already counted in the first term $(2^{\frac{n+1}{2}}-2)$. However, we will use a new assignment $\{re(f)[r-1]=0, re(f)[n-r+1]=0\}$, $\{re(f)[r+2]=0, re(f)[n-r-2]=0\}$, $\{re(f)[r]=1, re(f)[n-r]=1\}$, $\{re(f)[r+1]=1, re(f)[n-r-1]=1\}$ which again provides a balanced function as $\binom{n}{r-1}+\binom{n}{r+2}=\binom{n}{r}+\binom{n}{r+1}$. Note that this function is not counted in the first term as it selects both the terms from $\{\binom{n}{r}, \binom{n}{n-r}\}$, $\{\binom{n}{r+1}, \binom{n}{n-r-1}\}$ and does not select any term from $\{\binom{n}{r-1}, \binom{n}{n-r+1}\}$, $\{\binom{n}{r+2}, \binom{n}{n-r-2}\}$. For such a function f, the function f^c is also balanced and is not counted in the first term for the same reason. Thus we get the contribution $2 \times 2^{\frac{n+1}{2}-4} = 2^{\frac{n+1}{2}-3}$. ■

The functions constructed in this theorem are apart from the $2^{\frac{n+1}{2}}$ functions reported in [11, 80]. Theorem 3.5.2 justifies why Brüer [11] got an exception for $n = 13$. This will follow for $n = 33, 61, \ldots$ and so on. Note that, there exists some other classes of nontrivial examples which are not covered by Theorem 3.5.1, 3.5.2. As example for $n = 24$ we find $| A_n(1,5) | = 50$ and for $n = 31$, $| A_n(1,5) | = 66176$. However, the patterns do not provide simple relations as in Theorem 3.5.1, 3.5.2.

3.6 Correlation Immunity

Here we consider the characterization and construction problems for the set of symmetric correlation immune functions. We first provide an important characterization of symmetric correlation immune functions and then translate this characterization into a condition on binomial coefficients. From this we provide new classes of correlation immune functions.

Symmetric functions with both balancedness and correlation immunity properties have been considered in [40]. We here consider only the correlation immunity property for the symmetric Boolean functions. Let us now briefly describe the results on symmetric correlation immune functions provided in [118]. Theorem 8 of [118] provides a formula for $| A_n(4,5) |$. Note that, the reported equality of Theorem [118, Theorem 8] is only a lower bound. The result in [118, Theorem 8] says *If* $n > 1$, *then* $| A_n(4,5) | = 2^{\lfloor \frac{n}{2} \rfloor + 1}$ *where* $\lfloor x \rfloor$ *denotes the integer part of* x. The proof of this theorem in [118] uses the idea that *all functions with the conditions C4 and C5 satisfy* $f(X_1, X_2, \ldots, X_n) = f(1 \oplus X_1, 1 \oplus X_2, \ldots, 1 \oplus X_n)$, i.e., the output column of f will be palindromic. However, this is not always true and the stated condition is only a sufficient condition. As a counterexample, consider $f(X_1, X_2, X_3) = X_1 \oplus X_2 \oplus X_3 \in A_3(4,5)$, where $f(0,0,0) \neq f(1,1,1)$. In general for odd n, the two nondegenerate affine functions are nonpalindromic and in $A_n(4,5)$. So the exact statement of the theorem should be, $| A_n(4,5) | \geq 2^{\lfloor \frac{n}{2} \rfloor + 1}$. Also it should be noted that the result $| A_n(2,4,5) | = | A_n(4,5) | - 2$ in [118, Theorem 9(vi)] should be modified to $| A_n(2,4,5) | = | A_n(4,5) | - 4$, as there are exactly 4 affine functions in $A_n(4,5)$.

First we prove an important characterization of $A_n(4,5)$, the set of symmetric correlation

immune Boolean functions. For the purpose of construction it is convenient to translate this into a condition on binomial coefficients. Let $WTS(f) = \{i \mid re(f)[i] = 1\}$.

Theorem 3.6.1 *Let $f \in A_n(5)$ with $WTS(f) = \{i_1, \ldots, i_r\}$. Then f is CI iff $\binom{n-1}{i_1} + \ldots + \binom{n-1}{i_r} = \binom{n-1}{i_1-1} + \ldots + \binom{n-1}{i_r-1}$.*

Proof : Let $f = f_1 f_2$, with $f_1, f_2 \in A_{n-1}(5)$. We first show f is CI iff $wt(f_1) = wt(f_2)$. If f is CI then the condition is clearly necessary. Conversely suppose that $wt(f_1) = wt(f_2)$, and let X_i be any variable. Apply a permutation to the variables $X_1, \ldots, X_n$ such that X_i becomes the variable at the leftmost column in the truth table. Since f is symmetric, the output column in the truth table of f remains the same irrespective of such a permutation and hence the resulting function is again $f = f_1 f_2$. Thus, $Prob(f = X_i) = \frac{1}{2}$. This proves that f is CI. The condition in the theorem is equivalent to $wt(f_1) = wt(f_2)$. ■

As example, we have $\binom{7}{3} + \binom{7}{6} = \binom{7}{2} + \binom{7}{5}$. If we consider $f \in A_8(5)$, with $WTS(f) = \{3, 6\}$, then f is correlation immune. If n is even, then all the functions, we construct next, are in the set $A_n(2, 3, 4, 5)$, and if n is odd then all the functions, except the two affine ones, are in the set $A_n(2, 3, 4, 5)$. First we describe a sufficient condition for a function to be in $A_n(4, 5)$.

Theorem 3.6.2 *Consider integers n, r, i such that $2\binom{n-1}{r} = \binom{n-1}{r-i} + \binom{n-1}{r+i}$ for $i \geq 1$. Then one can construct a nonpalindromic function $f \in A_n(4, 5)$.*

Proof : Let us consider $2\binom{n-1}{r} = \binom{n-1}{r-i} + \binom{n-1}{r+i}$ for $i \geq 1$. This implies
$\binom{n-1}{r} + \binom{n-1}{n-r-1} = \binom{n-1}{r-i} + \binom{n-1}{n-r-i-1}$. Now we add
$(\binom{n-1}{r-1} + \ldots + \binom{n-1}{r-i+1}) + (\binom{n-1}{n-r-2} + \ldots + \binom{n-1}{n-r-i})$ to both sides. This gives
$(\binom{n-1}{r} + \binom{n-1}{r-1} + \ldots + \binom{n-1}{r-i+1}) + (\binom{n-1}{n-r-1} + \binom{n-1}{n-r-2} + \ldots + \binom{n-1}{n-r-i}) = (\binom{n-1}{r-1} + \binom{n-1}{r-2} + \ldots + \binom{n-1}{r-i}) + (\binom{n-1}{n-r-2} + \binom{n-1}{n-r-3} + \ldots + \binom{n-1}{n-r-i-1})$.

Now let us consider a function $f \in A_n$ such that $WTS(f) = \{r, r-1, \ldots, r-i+1, n-r-1, n-r-2, \ldots, n-r-i\}$. Thus, from Theorem 3.6.1, f is CI. For each of the input vectors of weights $r, r-1, \ldots, r-i+1, n-r-1, n-r-2, \ldots, n-r-i$, the output of function f is 1. This proves that the function is nonpalindromic. ■

Based on the sufficient condition of Theorem 3.6.1, we provide a generic construction method for functions in $A_n(4, 5)$.

Proposition 3.6.1 *(1) Let $f \in A_n(4, 5)$ with $re(f)[k] = re(f)[n-k] = 0$. Let $f' \in A_n(5)$ be such that $re(f')[i] = re(f)[i]$ for all i, $i \neq k, n-k$ and $re(f')[k] = re(f')[n-k] = 1$. Then $f' \in A_n(4, 5)$.*

(2) Let n be even and $f \in A_n(4,5)$ with $re(f)[\frac{n}{2}] = 0$. Let $f' \in A_n(5)$ be such that $re(f')[i] = re(f)[i]$ for all i, $i \neq \frac{n}{2}$ and $re(f')[\frac{n}{2}] = 1$. Then $f' \in A_n(4,5)$.

Proof : (1) Consider the pair of positions $\{\binom{n}{k}, \binom{n}{n-k}\}$ in $re(f)$. If $re(f)[k] = re(f)[n-k] = 0$, they do not contribute in the top half or the bottom half of the function. Now $re(f')[k] = re(f')[n-k] = 1$. Since $\binom{n-1}{k-1} + \binom{n-1}{n-k-1} = \binom{n-1}{k} + \binom{n-1}{n-k}$, they contribute the same number of 1's in the top and bottom half of the function f'. Thus if f is CI, then f' is CI too.

(2) We now consider the special case with even n. Here $re(f)[\frac{n}{2}] = 0$. Also, $\binom{n-1}{\frac{n}{2}} = \binom{n-1}{\frac{n}{2}-1}$. Thus we get the result. ■

Next we use the generic construction method to provide lower bounds. Note that for the following we use the notation $\mu(n)$, which takes the value 2 for odd n and the value 4 for even n.

Lemma 3.6.1 *Let $n+1$ be a perfect square and $n \geq 8$. Then $| A_n(2,3,4,5) | \geq 2^{\lfloor \frac{n}{2} \rfloor + 1} + 2^{\lceil \frac{n-1}{2} \rceil} - \mu(n)$.*

Proof : The first term provides the number of palindromic functions. We show that there are at least $2^{\lceil \frac{n-1}{2} \rceil}$ nondegenerate, nonaffine, nonpalindromic functions. When n is even, all the 4 symmetric affine functions are palindromic. When n is odd, only the degenerate affine functions are palindromic. This explains the subtraction of $\mu(n)$.

As given in Theorem 3.6.2, if we take $i = 1$, then $2\binom{n-1}{r} = \binom{n-1}{r-1} + \binom{n-1}{r+1}$. From this condition we get, $(n-2r-1)^2 = n+1$. The condition can also be written as, $\binom{n-1}{r} + \binom{n-1}{n-r-1} = \binom{n-1}{r+1} + \binom{n-1}{n-r}$. We construct a function $f \in A_n(5)$, where $re(f)[r+1] = 1$ and $re(f)[n-r] = 1$. We also fix $re(f)[n-r-1] = 0$ and $re(f)[r] = 0$. Thus f is nonpalindromic. Apart from these, consider the pair of positions $\{\binom{n}{i}, \binom{n}{n-i}\}$. If $re(f)[i] = re(f)[n-i] = 0$, they do not contribute in top half or bottom half of the function. If $re(f)[i] = re(f)[n-i] = 1$, then similar to Proposition 3.6.1(1), since $\binom{n-1}{i-1} + \binom{n-1}{n-i-1} = \binom{n-1}{i} + \binom{n-1}{n-i}$, they contribute the same number of 1's in the top and bottom half of the function. Moreover, similar to Proposition 3.6.1(2), for even n, $\binom{n-1}{\frac{n}{2}} = \binom{n-1}{\frac{n}{2}-1}$.

Hence for each of the $\lceil \frac{n-3}{2} \rceil$ different pair of positions, there exists 2 options (either both 0 or both 1) for constructing a nonpalindromic CI function f. Thus we get $2^{\lceil \frac{n-3}{2} \rceil}$ different functions. Also we have f is CI iff f^c is CI. Thus we get $2 \times 2^{\lceil \frac{n-3}{2} \rceil} = 2^{\lceil \frac{n-1}{2} \rceil}$ distinct nonpalindromic functions. ■

We get such functions for $(n = 8, r = 2), (n = 15, r = 5), \ldots$ etc.

Lemma 3.6.2 *Let $n+3$ be a perfect square and $n \geq 13$. Then* $| A_n(2,3,4,5) | \geq 2^{\lfloor \frac{n}{2} \rfloor + 1} + 2^{\lceil \frac{n-5}{2} \rceil} - \mu(n)$.

Proof : Taking $i = 2$ in Theorem 3.6.2, we get $2\binom{n-1}{r} = \binom{n-1}{r-2} + \binom{n-1}{r+2}$. From this condition we have, $(n-2r-1)^2 = n+3$. We also get, $\binom{n-1}{r+1} + \binom{n-1}{r} + \binom{n-1}{n-r} + \binom{n-1}{n-r-1} = \binom{n-1}{r+2} + \binom{n-1}{r+1} + \binom{n-1}{n-r+1} + \binom{n-1}{n-r}$. Thus we initially choose 4 positions and correspondingly fix the four palindromic positions to make $re(f)$ nonpalindromic. Similar to Lemma 3.6.1 we get $2 \times 2^{\lceil \frac{n-7}{2} \rceil} = 2^{\lceil \frac{n-5}{2} \rceil}$ distinct nonpalindromic functions. ∎

We get such functions for $(n = 13, r = 4), (n = 22, r = 8), \ldots$ etc. There may also be some other types of constructions in the same line.

Lemma 3.6.3 *Let $n+2$ be a perfect square and $n \geq 14$. Then* $| A_n(2,3,4,5) | \geq 2^{\lfloor \frac{n}{2} \rfloor + 1} + 2^{\lceil \frac{n-1}{2} \rceil} - \mu(n)$.

Proof : We take $f \in A_n(5)$ such that $WTS(f) = \{r+2, n-r\}$ with the condition $\binom{n-1}{r+2} + \binom{n-1}{n-r} = \binom{n-1}{r+1} + \binom{n-1}{n-r-1}$. Simplifying this condition, we get, $(n-2r-2)^2 = n+2$. We fix two pairs of positions to construct the nonpalindromic function. Now similar to Lemma 3.6.1 we get $2 \times 2^{\lceil \frac{n-3}{2} \rceil} = 2^{\lceil \frac{n-1}{2} \rceil}$ distinct nonpalindromic functions. ∎

Thus, we get such functions for $(n = 14, r = 4), (n = 23, r = 8), \ldots$ etc. From Lemma 3.6.1, Lemma 3.6.2 and Lemma 3.6.3, we get an improved lower bound on $| A_n(2,3,4,5) |$ for $n = \pi - i$, where $i = 1, 2, 3$ and π a perfect square, with $\pi \geq 16$.

Theorem 3.6.3 *Let π be a perfect square, with $\pi \geq 16$. Then it is possible to construct nonpalindromic $f \in A_n(2,3,4,5)$ for $n = \pi - i$, where $i = 1, 2, 3$.*

We can also consider the following theorem which can be proved in the same line of Theorem 3.6.2.

Theorem 3.6.4 *Consider integers n, r, i such that $2\binom{n-1}{r} = \binom{n-1}{r-i-1} + \binom{n-1}{r+i}$ for $i \geq 1$. Then there exists nonpalindromic $f \in A_n(4,5)$.*

Taking $i = 1$, we get the following corollary.

Corollary 3.6.1 *Consider integers n, r such that $2\binom{n-1}{r} = \binom{n-1}{r-2} + \binom{n-1}{r+1}$. Then one can construct nonpalindromic $f \in A_n(4,5)$.*

Note that, the condition $2\binom{n-1}{r} = \binom{n-1}{r-2} + \binom{n-1}{r+1}$ does not provide any simple Diophantine equation as in Lemma 3.6.1. However, running *Mathematica* [117] software, we could obtain

n	3	4	5	6	7	8	9	10	11	12	13	14
$\mid A_n(4,5) \mid$	6	8	10	20	26	48	42	64	66	144	178	452

15	16	17	18	19	20	21	22	23	24	25	26
428	576	514	1072	1442	2864	2534	4608	6402	12448	9350	16648

27	28	29	30	31	32	33
16522	32768	36866	82496	77186	132352	148170

Table 3.1: Enumeration of $A_n(4,5)$

two solutions $(n = 8, r = 5)$ and $(n = 20, r = 6)$. This will provide better lower bounds on $\mid A_8(2,3,4,5) \mid$ and $\mid A_{20}(2,3,4,5) \mid$.

Next we provide exact enumeration of $A_n(4,5)$ up to $n = 33$ which is listed in the Table 3.1. The values are obtained by running a computer program. Note that the bounds described in Lemma 3.6.1, Lemma 3.6.2 and Lemma 3.6.3 provide very close bounds with respect to what mentioned in the table for $\pi = 25$ and $n = 22, 23, 24$. In fact, for $n = 22$ (Lemma 3.6.2), we get the value 4604, which added with the four symmetric affine functions provides the exact count 4608 (in the table). For $n = 23$ (Lemma 3.6.3), the value is $6142 + 2$ compared to 6402 (in the table) and for $n = 24$ (Lemma 3.6.1), the value is $12284 + 4$ compared to 12448 (in the table). It is interesting to note that $\mid A_n(4,5) \mid = 2^{\lfloor \frac{n}{2} \rfloor + 1} + 2(n \bmod 2)$ for $n = 4, 5, 10, 11, 17, 28$. However, n does not follow any obvious pattern for the exact equality condition.

Chapter 4

Balanced Boolean Functions

Three basic properties of Boolean functions to be useful for symmetric key cryptosystems are balancedness, high algebraic degree and high nonlinearity. In addition, strict avalanche criteria and propagation characteristics are required for design of S-boxes while correlation immunity is required for design of stream cipher systems. In this chapter a variety of new techniques are introduced to construct Boolean functions possessing one or more of the above properties. The techniques are powerful enough to solve several open problems and to improve significantly upon previous research. An important aspect of this work is to show new ways of using bent functions and the functions provided by Patterson and Wiedemann [88, 89] to construct previously unknown cryptographically important Boolean functions.

4.1 Introduction

Balancedness, high nonlinearity, and high algebraic degree are essential properties of Boolean functions for use in both stream and block ciphers. An S-box in a block cipher is a set of Boolean functions. Each of the constituent Boolean functions must satisfy *strict avalanche criteria and propagation characteristics* to resist cryptanalytic attacks. On the other hand, in stream cipher systems a Boolean function is used to combine the outputs of several independent sequences produced by Linear Feedback Shift Registers (LFSRs). Such a Boolean function must be *correlation immune* to resist certain types of divide and conquer attacks.

Here we consider Boolean functions possessing one or more of the above properties. We provide new construction methods for such functions, solve several open problems and significantly improve upon existing results. Patterson and Wiedemann [88, 89] have provided an important class of highly nonlinear functions. We investigate the structure of these functions and provide new ways to use these functions for the construction of highly nonlinear

balanced functions (Section 4.2) and highly nonlinear balanced functions satisfying SAC(k) with maximum algebraic degree (Section 4.4). Also we show new ways of applying bent functions for the construction of highly nonlinear functions satisfying important cryptographic properties.

It is known [95] that for even n, the maximum nonlinearity achievable by a Boolean function is $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$. Such functions are called bent functions and their combinatorial properties have been studied [95, 93, 31, 32, 17, 16]. These functions are not balanced. Moreover, for $n \geq 4$ the algebraic degree of a bent function is at most $\frac{n}{2}$ [95]. Construction of highly nonlinear balanced Boolean functions on even number of variables has been proposed in [104, 32] by modifying the bent functions. For odd n, the corresponding class of functions with maximum nonlinearity has not been characterized. For odd $n \leq 7$, the maximum possible nonlinearity is known [83] and the value is $2^{n-1} - 2^{\frac{n-1}{2}}$. For $n = 9, 11, 13$ functions with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ can be constructed, though it is not known whether this is the maximum possible nonlinearity. For odd $n \geq 15$, it is known [88, 89] that there exists functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$. However, these functions are not balanced. Balanced functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ has been constructed for odd $n \geq 29$ [104]. Construction of highly nonlinear balanced functions has also been considered in [18, 77, 78, 79, 86]. The algebraic degree of highly nonlinear balanced functions for $n \leq 9$ has been considered in [33, 34].

We use a randomized heuristic to construct for the first time balanced functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for $n = 15, 17, 19, 21, 23, 25, 27$. We use the functions provided in [89] as the basic input to our algorithm. Earlier these functions [89] were used to obtain balanced functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ only for odd $n \geq 29$ [104]. We also provide a recursive algorithm for obtaining currently best known nonlinearity for balanced Boolean functions. Further, for odd $n \geq 3$, we provide construction methods for balanced functions with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ having algebraic degree both $(n-2)$ and $(n-1)$. In [33], balanced functions of 7 variables with nonlinearity 56, algebraic degree 6 and balanced functions of 9 variables with nonlinearity 240, algebraic degree 7 has been reported. The method needs an exhaustive search over a subclass of Boolean functions called the idempotents and the search method is infeasible for more number of variables. Our result generalizes the results of [33].

Walsh Transform is a powerful tool in the analysis of Boolean functions. It is important to note that (see Chapter 3) $F(\overline{\omega}) = wd(f, \oplus_{i=1}^{i=n} \omega_i X_i)$, where F is the Walsh transform of f. Correlation immune functions were introduced by Siegenthaler [109, 110]. Later Xiao and Massey provided another characterization of such functions in [42]. Here we use this characterization as a definition.

Definition 4.1.1 *A function $f(X_1, X_2, \ldots, X_n)$ is mth order correlation immune iff its Walsh transform F satisfies $F(\overline{\omega}) = 0$, for $1 \leq wt(\overline{\omega}) \leq m$. A balanced m-th order cor-*

relation immune function is called m-resilient.

Siegenthaler [109] proved that an n-input, m-resilient function with algebraic degree d satisfies the inequality $m + d \leq n - 1$. Functions for which $m + d = n - 1$ is called optimized with respect to Siegenthaler's inequality (in short *optimized*). In this chapter we will mainly consider the optimized 1-resilient functions. The discussion on construction of m-resilient functions will be elaborated in Chapter 7.

In [86], it was conjectured that the maximum nonlinearity of 1-resilient functions on even number of variables n can be at most $2^{n-1} - 2^{\frac{n}{2}}$. We disprove this conjecture for all even $n \geq 16$ by showing that one can construct 1-resilient functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n}{2}}$. We provide construction methods for n-variable, 1-resilient optimized functions having nonlinearity $2^{n-1} - 2^{\lfloor \frac{n}{2} \rfloor}$. The method uses linear algebraic techniques. As example, using this one can construct a 9-variable optimized 1-resilient function with nonlinearity 240, which was not known earlier. Such functions can be efficiently used to construct m-resilient functions (see Chapter 7).

S-boxes can be viewed as a set of Boolean functions [93, 92, 58]. Propagation Characteristic(PC) and Strict Avalanche Criteria(SAC) are important properties of Boolean functions to be used in S-boxes. Preneel et al [93, 92] provided basic construction techniques for Boolean functions with these properties. In [93], it has been shown that for balanced SAC(k) functions on n variables, $deg(f) \leq n - k - 1$. Recently in [58], balanced SAC(k) functions on n variables with $deg(f) = n - k - 1$ has been identified for $n - k - 1 =$ odd. However, construction of such functions for $n - k - 1 =$ even has been left as an open problem. In [58] balanced SAC(k) functions with high algebraic degree and in [103] highly nonlinear balanced SAC functions have been proposed. However, balanced SAC(k) functions with both high algebraic degree and high nonlinearity have not been studied. PC(l) of order k functions with good nonlinearity and algebraic degree have been reported in [58]. These properties have also been considered in [35, 105, 104, 106, 57, 119, 44].

We use a linear transformation on the input variables of the functions provided in [89] to obtain PC(1) functions on n variables (for odd $n \geq 15$) with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^{\frac{n-15}{2}}$. This establishes the existence of PC(1) functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd $n \geq 15$, which was not known earlier. We improve the algebraic degree and nonlinearity results of the PC(l) of order k functions reported in [58]. Motivated by the construction methods of SAC(k) functions in [58], we introduce a new cryptographic criterion called the *restricted balancedness* of Boolean functions and show that certain types of bent functions satisfy this property. Also we modify the functions provided by Patterson and Wiedemann [88, 89] to obtain restricted balancedness while keeping the nonlinearity unchanged. This requires an in-depth study of the internal structure of the functions provided in [89]. For the first time we consider the properties of balancedness, SAC(k), algebraic degree and nonlinearity together. We construct balanced (using the functions with restricted

balancedness) SAC(k) functions in Ω_n with maximum possible algebraic degree $n-k-1$ and very high nonlinearity for $k \leq \frac{n}{2}-1$. This also shows that there exists balanced SAC(k) functions on n variables with $deg(f) = n-k-1 =$ even, which was posed as an open question in [58]. We also present an interesting result on resilient functions satisfying PC(k). In an previous work [105], it was shown that resilient functions satisfy propagation characteristics with respect to a set of input vectors, but not PC(k) for some k.

Before proceeding we state a few simple results which will be useful later.

Proposition 4.1.1 *Given $l_1, l_2 \in L(n)$, $d(l_1, l_2) = 0, 2^{n-1}, 2^n$ ($wd(l_1, l_2) = 2^n, 0, -2^n$) according as $l_1 = l_2, l_1 \neq l_2$ or $l_2^c, l_1 = l_2^c$.*

Proposition 4.1.2 *Let $f \in \Omega_n$ and $f = f_1 f_2$, where $f_1, f_2 \in \Omega_{n-1}$. If $wt(f)$ is odd then algebraic degree of f is n. Moreover, if both $wt(f_1)$ and $wt(f_2)$ are odd then the algebraic degree of f is $n-1$.*

Proposition 4.1.3 *Given a balanced function $f \in \Omega_n$ with $nl(f) = x$, one can construct balanced $f' \in \Omega_n$ with $nl(f') \geq x-2$ and $deg(f') = n-1$.*

Proof : Let $f = f_1 f_2$ where $f_1, f_2 \in \Omega_{n-1}$. If $wt(f_1)$ and $wt(f_2)$ are both odd, then $deg(f) = n-1$ and take $f' = f$. If we have both $wt(f_1)$ and $wt(f_2)$ even, toggle one bit of f_1 from 0 to 1 (get f_1') and one bit of f_2 from 1 to 0 (get f_2'). Thus we get a balanced function $f' = f_1' f_2' \in \Omega_n$, with both $wt(f_1')$ and $wt(f_2')$ odd, i.e. with algebraic degree $(n-1)$. The nonlinearity can fall by at most two for toggling two bits. ■

4.2 Nonlinearity of Balanced Functions

In this section we discuss the nonlinearity and algebraic degree for balanced functions. Patterson and Wiedemann [89] constructed 15-variable functions with nonlinearity 16276 and weight 16492. Seberry, Zhang and Zheng [104] used such functions to construct balanced functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd $n \geq 29$. More recently, Fontaine [34] studied the functions of [88, 89]. However, currently existing works do not balanced 15-variable functions with nonlinearity greater than 16256.

We identify an important result which is the first step towards constructing a balanced 15-variable function with nonlinearity greater than 16256.

Proposition 4.2.1 *It is possible to construct $f \in \Omega_{15}$ with nonlinearity 16276 and weight 16364.*

Proof : Consider a function $f_1 \in \Omega_{15}$ with $nl(f_1) = 16276$ and $wt(f_1) = 16492$. From [89], we know that there are 3255 linear functions in $L(15)$ at a distance 16364 from f_1. Let l be one of these 3255 linear functions. Define $f = f_1 \oplus l$. Then $f \in \Omega_{15}$ and $nl(f) = nl(f_1) = 16276$ and $wt(f) = wt(f_1 \oplus l) = d(f_1, l) = 16364$. ■

Here we consider the function $f_1 \in \Omega_{15}$, with $nl(f_1) = 16276$. The existence of the function f_1 has been shown by Patterson and Wiedemann [88, 89] by experiment, where they have reduced the search space using mathematical techniques. We have also run the same experiment to get such a function. Once one can construct such a function, the existence of the function is settled.

Next we have the following randomized heuristic for constructing highly nonlinear balanced functions for odd $n \geq 15$.

Algorithm 1 : RandBal(n)

1. Let f be a function constructed using Proposition 4.2.1. Let $n = 2k + 15$, $k \geq 0$ and let $F \in \Omega_n$ be defined as follows. For $k = 0$, take $F = f$, and for $k > 0$, take $F = f(X_1, \ldots, X_{15}) \oplus g(X_{16}, \ldots, X_n)$, where $g \in \Omega_{2k}$ is a bent function. Note that $nl(F) = 2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^k$ and $wt(F) = 2^{n-1} - 20 \times 2^k$.

2. Divide the string F in Ω_n into 20×2^k equal contiguous substrings, with the last substring longer than the rest.

3. In each substring choose a position with 0 value uniformly at random and change that to 1. This generates a balanced function $F_b \in \Omega_n$.

4. If $nl(F_b) > 2^{n-1} - 2^{\frac{n-1}{2}}$, then report. Go to step 1 and continue.

We have run this experiment number of times and succeeded in obtaining plenty of balanced functions with nonlinearities $2^{14} - 2^7 + 6$, $2^{16} - 2^8 + 18$, $2^{18} - 2^9 + 46$ and $2^{20} - 2^{10} + 104$ respectively for 15, 17, 19 and 21 variables. It is possible to distribute the 0's and 1's in the function in a manner (changing step 2, 3 in Algorithm 1) such that weight of the upper and lower half of the function are odd. This provides balanced functions with maximum algebraic degree $(n-1)$ and the same nonlinearity as before. Note that, running Algorithm 1 for large n is time consuming. However, we can extend the experimental results in a way similar to that in [88]. Consider a bent function $g(Y_1, \ldots, Y_{2k}) \in \Omega_{2k}$ and $f(X_1, \ldots, X_{21})$ with nonlinearity $2^{20} - 2^{10} + 104$ as obtained from Algorithm RandBal(). Let $h \in \Omega_{21+2k}$ such that $h = g \oplus f$. Then we get $nl(h) = 2^{20+2k} - 2^{10+k} + 104 \times 2^k$. These functions can be modified to get algebraic degree $(n-1)$ as in Proposition 4.1.3. Thus we get the following result.

Theorem 4.2.1 *One can construct balanced Boolean functions on* $n = 15 + 2k$ $(k \geq 0)$ *variables with nonlinearity greater than* $2^{n-1} - 2^{\frac{n-1}{2}}$. *Moreover, such functions can have algebraic degree* $(n-1)$.

Dobbertin [32] provided a recursive procedure for modifying a general class of bent functions to obtain highly nonlinear balanced Boolean functions on even number of variables. A special case of this procedure which modifies Maiorana-McFarland class of bent functions was provided in [104]. For even n, it is conjectured in [32] that the maximum value of nonlinearity of balanced functions, which we denote by $nlbmax()$, satisfies the recurrence: $nlbmax(n) = 2^{n-1} - 2^{\frac{n}{2}} + nlbmax(\frac{n}{2})$.

We next provide a combined interlinked recursive algorithm to construct highly nonlinear balanced functions for both odd and even n. Note that for even number of variables, Algorithm 2 uses a special case of the recursive construction in [32]. Further we show how to obtain maximum algebraic degree. The input to this algorithm is n and the output is balanced $f \in \Omega_n$ with currently best known nonlinearity.

Algorithm 2 : BalConstruct(n)

1. If n is odd

   (a) if $3 \leq n \leq 13$ construct f with algebraic degree $(n-1)$ and nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ (see Theorem 4.2.3).

   (b) if $15 \leq n \leq 21$ return f to be the best function constructed by RandBal(n).

   (c) if $n \geq 23$

      i. Let $h_1 \in \Omega_{n-21}$ be bent and $g_1 \in \Omega_{21}$ be the best nonlinear function constructed by RandBal(n). Let $f_1 \in \Omega_n$ be such that $f_1 = h_1 \oplus g_1$.

      ii. Let $h_2 =$ BalConstruct$(n-15)$ and $g_2 \in \Omega_{15}$ as in Proposition 4.2.1. Let $f_2 \in \Omega_n$ be such that $f_2 = h_2 \oplus g_2$.

      iii. If $nl(f_1) \geq nl(f_2)$ return f_1 else return f_2.

2. If n is even

   Let $h =$ BalConstruct$(\frac{n}{2})$. Let f be the concatenation of h followed by $2^{\frac{n}{2}} - 1$ distinct nonconstant linear functions on $\frac{n}{2}$ variables. Return f.

To obtain maximum algebraic degree in the above algorithm we need the following modifications.

- For odd $n \leq 13$, the functions available from Proposition 4.1.1 guarantees degree $(n-1)$.

- For odd n, $15 \leq n \leq 21$, modification of algorithm RandBal() guarantees algebraic degree $(n-1)$ without dropping nonlinearity.
- For odd $n \geq 23$, using Proposition 4.1.3, degree $(n-1)$ can be achieved sacrificing nonlinearity by at most 2.
- For even n, recursively ensure that algebraic degree of h (in Step 2 of BalConstruct()) is $\frac{n}{2} - 1$.

Let $nlb(n)$ be the nonlinearity of an n-variable balanced function constructed by BalConstruct(n). Similarly let $nla(n)$ be the nonlinearity of an n-variable balanced function with degree $n-1$, constructed by Algorithm BalConstruct(n). We have the following performance guarantee on $nlb(n)$ and $nla(n)$.

Theorem 4.2.2 *Algorithm BalConstruct(n) constructs a balanced n-variable function having nonlinearity $nlb(n)$, given by,*

$$
\begin{aligned}
nlb(n) &= 2^{n-1} - 2^{\frac{n-1}{2}} && \text{for odd } n,\ 3 \leq n \leq 13, && (1)\\
&= \sigma_{15} + 6,\ \sigma_{17} + 18,\ \sigma_{19} + 46,\ \sigma_{21} + 104 && \text{for } n = 15, 17, 19, 21, && (2)\\
&= \max(A, B) && \text{for odd } n \geq 23, && (3)\\
&= 2^{n-1} - 2^{\frac{n}{2}} + nlb(\tfrac{n}{2}) && \text{for even } n && (4)
\end{aligned}
$$

where $k = \frac{n-15}{2}$, $\sigma_n = 2^{n-1} - 2^{\frac{n-1}{2}}$,
$A = 2^{n-1} - 2^{\frac{n-1}{2}} - 88 \times 2^k + 216 \times nlb(k)$ *and* $B = 2^{n-1} - 2^{\frac{n-1}{2}} + 13 \times 2^k$.
For cases (1) and (2), $nla(n) = nlb(n)$. *For case (3),* $nla(n) \geq nlb(n) - 2$ *and for case (4)* $nla(n) = 2^{n-1} - 2^{\frac{n}{2}} + nla(\frac{n}{2})$.

It should be noted that, from Theorem 4.2.2, the nonlinearity measures $nla()$ and $nlb()$ satisfy the same recurrence of $nlbmax()$ as proposed by Dobbertin [32]. Algorithm BalConstruct(n) provides currently best known nonlinear balanced functions for all n. In particular, we summarize the following points to highlight the performance of the algorithm.

1. Our method provides balanced functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$ for odd n, $15 \leq n \leq 27$ which were not known earlier.
2. For even n, the nonlinearity obtained by our method is strictly greater than that mentioned in [104, Theorem 13] for all $n = 2^s(2t+1),\ s \geq 1, t \geq 7$.
3. For odd n the nonlinearity obtained by our method is strictly greater than that mentioned in [104] for all $n = 2^s(2t+1) + 15,\ s \geq 1, t \geq 7$.
4. Apart from points 2 and 3 above, we also obtain strictly better nonlinearities for other values of n. For comparison consider pairs of the form, **<no of variables, our**

nonlinearity - nonlinearity of [104]>. Then we get $< 29,832 >$, $< 31,800 >$ as examples of better performance.

In this section we have shown how to heuristically modify the Patterson-Wiedemann functions to obtain balancedness while retaining nonlinearity higher than the bent concatenation bound. However, the question of mathematically constructing such functions remains open. Also settling the conjecture in [32] is an important unsolved question.

4.2.1 Algebraic Degree

Here we present two results on the construction of n-variable (n odd) balanced functions with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ and algebraic degree $(n-1)$ and $(n-2)$. For odd $n \leq 7$, this provides the best possible nonlinearity [83] and for $n = 9, 11, 13$, this provides the best known nonlinearity. For odd $n \geq 15$, the constructions of Theorem 4.2.2 yield better nonlinearities for balanced functions with degree $(n-1)$. However, the results obtained here are required in the construction of optimized resilient functions with best known nonlinearities (see Section 4.3 later).

Using exhaustive search over idempotents, Filiol and Fontaine [33] reported 7-variable balanced functions with nonlinearity 56, algebraic degrees 5, 6 and 9-variable balanced functions with nonlinearity 240, algebraic degree 7. However, it is infeasible to perform such a search for large n. Moreover, 9-variable balanced function with nonlinearity 240 and algebraic degree 8 was not obtained in [33]. Theorem 4.2.3 provides a general solution to this problem.

Theorem 4.2.3 *It is possible to construct balanced $f \in \Omega_n$, $n \geq 3$ odd, with algebraic degree $(n-1)$ and nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$.*

Proof : For $n = 3$, $f = X_3 \oplus X_1X_2$ satisfies the condition. Now we consider the case for $n \geq 5$. We take bent functions $h_1, h_2 \in \Omega_{n-1}$ of the following form. The function h_1 is a concatenation of $2^{\frac{n-1}{2}}$ distinct linear functions of $\frac{n-1}{2}$ variables where the first linear function is the all zero function. The function h_2 is same as h_1 except the first function is the all one function in place of the all zero function. Thus, the binary strings are of the form $h_1 = 0^{2^{\frac{n-1}{2}}}\lambda$ and $h_2 = 1^{2^{\frac{n-1}{2}}}\lambda$, where λ is a binary string of length $2^{n-1} - 2^{\frac{n-1}{2}}$, the concatenation of $2^{\frac{n-1}{2}} - 1$ distinct nonconstant linear functions of $\frac{n-1}{2}$ variables. Note that 0^x (resp. 1^x) denotes the all zero (resp. all one) string of length x. Next we construct $h_1' = 10^{2^{\frac{n-1}{2}}-1}\lambda$ and $h_2' = 01^{2^{\frac{n-1}{2}}-1}\lambda$ by toggling the first bit of both h_1, h_2. We define $f = (1 \oplus X_n)h_1' \oplus X_n h_2'$, i.e. $f = h_1'h_2'$, the concatenation of strings h_1' and h_2'.

Note that $wt(f) = wt(h_1') + wt(h_2') = 2^{n-1}$. Thus f is balanced.
Since, both $wt(h_1'), wt(h_2')$ are odd, f is of algebraic degree $(n-1)$. Next we calculate the

nonlinearity of f.

CASE 1: We first consider affine functions $ll \in L(n)$, where $l \in L(n-1)$. We write $l = l_x l_y$, where l_x is a binary string of length $2^{\frac{n-1}{2}}$ and l_y is a binary string of length $2^{n-1} - 2^{\frac{n-1}{2}}$. Now, $d(f, ll) = d(10^{2^{\frac{n-1}{2}}-1}, l_x) + d(\lambda, l_y) + d(01^{2^{\frac{n-1}{2}}-1}, l_x) + d(\lambda, l_y)$. Since, $10^{2^{\frac{n-1}{2}}-1}$ and $01^{2^{\frac{n-1}{2}}-1}$ are bitwise complements, $d(10^{2^{\frac{n-1}{2}}-1}, l_x) + d(01^{2^{\frac{n-1}{2}}-1}, l_x) = 2^{\frac{n-1}{2}}$. Now consider $d(\lambda, l_y)$. We represent $\lambda = \lambda_1 \lambda_2 \ldots \lambda_p$ and $l_y = l_1 l_2 \ldots l_p$, $p = 2^{\frac{n-1}{2}} - 1$ and $\lambda_i, l_i \in L(\frac{n-1}{2})$. Three conditions may arise.

(1) $l_i = \lambda_i$ for some i, $1 \leq i \leq p$. Then $l_j \neq \lambda_j, l_j \neq \lambda_j^c$ for all $j \neq i$, $1 \leq j \leq p$. So, $d(\lambda, l_y) = (2^{\frac{n-1}{2}} - 2) \times 2^{\frac{n-1}{2}-1}$, i.e. $2d(\lambda, l_y) = 2^{n-1} - 2 \times 2^{\frac{n-1}{2}}$. Thus, $d(f, ll) = 2^{\frac{n-1}{2}} + 2^{n-1} - 2 \times 2^{\frac{n-1}{2}} = 2^{n-1} - 2^{\frac{n-1}{2}}$.

(2) $l_i = \lambda_i^c$ for some i, $1 \leq i \leq p$. Again $l_j \neq \lambda_j, l_j \neq \lambda_j^c$ for all $j \neq i$, $1 \leq j \leq p$.
Then $d(\lambda, l_y) = (2^{\frac{n-1}{2}} - 2) \times 2^{\frac{n-1}{2}-1} + 2^{\frac{n-1}{2}}$, i.e. $2d(\lambda, l_y) = 2^{n-1} - 2 \times 2^{\frac{n-1}{2}} + 2 \times 2^{\frac{n-1}{2}}$. Thus, $d(f, ll) = 2^{n-1} + 2^{\frac{n-1}{2}}$.

(3) $l_j \neq \lambda_j, l_j \neq \lambda_j^c$ for all $1 \leq j \leq p$.
Then $d(\lambda, l_y) = (2^{\frac{n-1}{2}} - 1) \times 2^{\frac{n-1}{2}-1}$, i.e. $2d(\lambda, l_y) = 2^{n-1} - 2^{\frac{n-1}{2}}$. Thus, $d(f, ll) = 2^{n-1}$.

CASE 2: Next we consider affine function $ll^c \in L(n)$, where $l \in L(n-1)$. Now, $d(f, ll^c) = d(10^{2^{\frac{n-1}{2}}-1}, l_x) + d(\lambda, l_y) + d(01^{2^{\frac{n-1}{2}}-1}, l_x^c) + d(\lambda, l_y^c)$. Note that $l_x \in L(\frac{n-1}{2})$. Since, $10^{\frac{n-1}{2}-1}$ and $01^{2^{\frac{n-1}{2}}-1}$ are bitwise complements, $d(10^{2^{\frac{n-1}{2}}-1}, l_x) + d(01^{2^{\frac{n-1}{2}}-1}, l_x^c) = 2 \times d(10^{2^{\frac{n-1}{2}}-1}, l_x) =$ 2 or $2^{\frac{n+1}{2}} - 2$ or $2^{\frac{n-1}{2}} \pm 2$. Also, $d(\lambda, l_y) + d(\lambda, l_y^c) = 2^{n-1} - 2^{\frac{n-1}{2}}$. Hence, $d(f, ll^c) = 2^{n-1} - 2^{\frac{n-1}{2}} + 2$ or $2^{n-1} + 2^{\frac{n-1}{2}} - 2$ or $2^{n-1} \pm 2$.

Thus, $2^{n-1} - 2^{\frac{n-1}{2}} \leq d(f, ll), d(f, ll^c) \leq 2^{n-1} + 2^{\frac{n-1}{2}}$, and hence $nl(f) = 2^{n-1} - 2^{\frac{n-1}{2}}$. ■

Theorem 4.2.4 *It is possible to construct balanced $f \in \Omega_n$, n odd, with algebraic degree $(n-2)$ and nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$.*

Proof : We consider a bent function $h_1 \in \Omega_{n-1}$ where h_1 is of the form $h_a h_b h_c$, h_a is the all zero string of length $2^{\frac{n-1}{2}}$, h_b is of length $2^{n-1} - 2 \times 2^{\frac{n-1}{2}}$ which is concatenation of $2^{\frac{n-1}{2}} - 2$ distinct linear functions of $\frac{n-1}{2}$ variables and h_c in $L(\frac{n-1}{2})$ is the linear function which is nondegenerate on $\frac{n-1}{2}$ variables. We toggle the first and last bits of h_1, i.e. the first bit of h_a and the last bit of h_c. This gives a string h_1' in Ω_{n-1} which is of the form xyz, where x is transformed from h_a by toggling the first bit, y is h_b and z is transformed from h_c by toggling the last bit. We take $f = xyzx^c yz^c$.

Now, $wt(f) = (wt(x) + wt(x^c)) + (2 \times wt(y)) + (wt(z) + wt(z^c)) = (2^{\frac{n-1}{2}}) + (2^{n-1} - 2 \times 2^{\frac{n-1}{2}}) + (2^{\frac{n-1}{2}}) = 2^{n-1}$.

Note that $wt(f_1)$, $wt(f_2)$, $wt(f_3)$, $wt(f_4)$ are all odd and hence f is of algebraic degree at least $n-2$. The algebraic degree of f cannot be n, since $wt(f)$ is even. Moreover, by

expanding the algebraic normal form of f, it can be seen that all terms of degree $n-1$ vanish.

The nonlinearity can be shown to be $2^{n-1} - 2^{\frac{n-1}{2}}$. The proof is similar to that of Theorem 4.2.3. ■

Example 4.2.1 *First we give an example corresponding to Theorem 4.2.3. Take bent functions* $h_1, h_2 \in \Omega_4$, *where* $h_1 = 0000010100110110$ *and* $h_2 = 1111010100110110$. *Now* $h_1' = 1000010100110110$ *and* $h_2' = 0111010100110110$ *and* $f = h_1'h_2' = 1000010100110110011101010100110110$. *Algebraic degree of* f *is* $5-1=4$ *and nonlinearity* $2^{5-1} - 2^{\frac{5-1}{2}} = 12$.

Next we give an example corresponding to Theorem 4.2.4. Take bent function $h_1 \in \Omega_4$, *where* $h_1 = 0000010100110110$.
Take $x = 1000, y = 01010011, z = 0111$, *and* $h_1' = xyz$ *i.e. we toggle first and last bit of* h_1 *to get* h_1'. *Now,* $f = xyzx^cyz^c = 1000010100110111011101010011100 0$. *Algebraic degree of* f *is* $5-2=3$ *and nonlinearity* $2^{5-1} - 2^{\frac{5-1}{2}} = 12$.

4.3 Resilient Functions

We use the results of the previous section to construct nonlinear optimized m-resilient functions. First we disprove a conjecture which appeared in [86]. The conjecture states that the maximum possible nonlinearity for 1-resilient function on n variables, n even, is $2^{n-1} - 2^{\frac{n}{2}}$.

Theorem 4.3.1 *For all even* $n \geq 16$, *it is possible to construct* 1*-resilient function on* n *variables with nonlinearity greater than* $2^{n-1} - 2^{\frac{n}{2}}$.

Proof : Since $n \geq 16$ and even, $n-1$ is odd and is ≥ 15. From Theorem 4.2.1 we have balanced $f \in \Omega_{n-1}$ with nonlinearity greater than $2^{n-2} - 2^{\frac{n-2}{2}}$. Define $F_1 = X_n \oplus f$ and $F_2 = (1 \oplus X_n)f(X_1, \ldots, X_n) \oplus X_n f(1 \oplus X_1, \ldots, 1 \oplus X_n)$. Then both F_1 and F_2 are 1-resilient [12] with nonlinearity $2nl(f)$ which is greater than $2^{n-1} - 2^{\frac{n}{2}}$. ■

A similar result holds for optimized functions (see Theorem 4.3.2). We now provide constructions of n-variable, m-resilient nonlinear functions with degree $n-m-1$. We first show how to construct n-variable, 1-resilient functions with degree $n-2$. The case for even n is as follows.

Theorem 4.3.2 *For even* $n \geq 4$, *it is possible to construct* n*-variable,* 1*-resilient functions with degree* $n-2$ *and nonlinearity* $x = 2nla(n-1)$. *The value of* x *is equal to* $2^{n-1} - 2^{\frac{n}{2}}$ *for* $4 \leq n \leq 14$ *and is greater than* $2^{n-1} - 2^{\frac{n}{2}}$ *for* $n \geq 16$.

Proof : Let f be an $(n-1)$-variable function constructed by BalConstruct() modified to ensure maximum degree $n-2$. Then ff^c is an n-variable, 1-resilient function having degree $n-2$ and nonlinearity $2nla(n-1)$. The value of $nla()$ from Theorem 4.2.2 is used to obtain the expressions here. ■

For odd n, we use a different method. We first describe a technique which has been used in [86]. Given a balanced function $f \in \Omega_n$, we define $S_f = \{\overline{\omega} \in \{0,1\}^n \mid F(\overline{\omega}) = 0\}$, where F is the Walsh transform of f. If there exists n linearly independent vectors in S_f, then we can construct a nonsingular $n \times n$ matrix B_f whose rows are linearly independent vectors from S_f. Let, $C_f = B_f^{-1}$. Now if we construct a function $f'(\overline{X}) = f(C_f\overline{X})$, then f' is balanced, and both f', f *have the same nonlinearity and algebraic degree.* Moreover, $F'(\overline{\omega}) = 0$ for $0 \leq wt(\overline{\omega}) \leq 1$, where F' is the Walsh Transform of f'. This ensures that f' is 1-resilient.

Though the technique has been used in [86], they started with a random Boolean function and hence could not obtain a nonlinearity of $2^{n-1} - 2^{\frac{n-1}{2}}$. Here we start with an n-variable balanced function constructed by Theorem 4.2.4, having degree $(n-2)$ and nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$. Since these parameters are preserved by the linear transformation on the variables, we obtain 1-resilient, n-variable function with degree $(n-2)$ and nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$.

Let ϵ_i^n be an n-bit vector with ith $(1 \leq i \leq n)$ entry 1 and all other entries 0. For example $\epsilon_n^n = (1,0,\ldots,0)$ and $\epsilon_1^n = (0,\ldots,0,1)$.

For $n \equiv 3 \bmod 4$, define $r_1 = \epsilon_{\frac{n-1}{2}}^n \oplus \ldots \oplus \epsilon_1^n$, $r_i = \epsilon_{n+1-i}^n$, $2 \leq i \leq \frac{n+1}{2}$ and $r_i = \epsilon_n^n \oplus \bigoplus_{1 \leq j \leq \frac{n-1}{2}, j \neq n+1-i} \epsilon_j^n$, $\frac{n+1}{2} + 1 \leq i \leq n$.

For $n \equiv 1 \bmod 4$, define $r_1 = \epsilon_1^n \oplus \epsilon_2^n \oplus \epsilon_{n-1}^n \oplus \epsilon_n^n$, $r_i = \epsilon_{n+1-i}^n$, $2 \leq i \leq \frac{n+1}{2}$ and $r_i = \epsilon_n^n \oplus \epsilon_{n+1-i}^n$, $\frac{n+1}{2} + 1 \leq i \leq n$.

Let B_n be a matrix whose rows are $r_1, \ldots, r_n$.

Example 4.3.1 *In this example we provide B_7 and B_9.*

$$B_7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \text{ and } B_9 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The following result follows from the definition of B_n.

Proposition 4.3.1 *The matrix B_n is nonsingular. Moreover, for $n \equiv 3 \bmod 4$, B_n is symmetric and is its own inverse.*

Proposition 4.3.2 *For odd n, let f be an n-variable function constructed by Theorem 4.2.4. Then for each row r_i of B_n, $F(r_i) = 0$, where F is Walsh transform of f.*

Proof : Here we only show $F(r_1) = 0$ for $n \equiv 3 \bmod 4$, the other cases being similar. In this case r_1 represents the linear function which is nondegenerate on all the variables $X_1, \ldots, X_{\frac{n-1}{2}}$. Let this function be l. Then we can write $l = \lambda^{2^{\frac{n+1}{2}}}$ (the string λ concatenated $2^{\frac{n+1}{2}}$ times) where $\lambda \in L(\frac{n-1}{2})$ and is nondegenerate on all the variables $X_1, \ldots, X_{\frac{n-1}{2}}$. We show that $wd(f, l) = 0$. We write $f = f_1 f_2 \ldots f_p$, $p = 2^{\frac{n+1}{2}}$. For $i \neq 1, \frac{p}{2}, \frac{p}{2}+1, p$, $wd(f_i, \lambda) = 0$, since f_i is a linear function different from λ or λ^c. Therefore, $wd(f, l) = wd(f_1, \lambda) + wd(f_{\frac{p}{2}}, \lambda) + wd(f_{\frac{p}{2}+1}, \lambda) + wd(f_p, \lambda)$. Now, f is of the form $xyzx^cyz^c$ in the proof of Theorem 4.2.4. Hence, $f_1 = x$, $f_{\frac{p}{2}} = z$, $f_{\frac{p}{2}+1} = x^c$ and $f_p = z^c$ and so $wd(f, l) = 0$. ■

Using Proposition 4.3.1, Proposition 4.3.2 and Theorem 4.2.4, we get the following result.

Theorem 4.3.3 *For odd $n \geq 5$, it is possible to construct 1-resilient, degree $(n-2)$ functions in Ω_n with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$.*

These functions combined with the recursive methods of Chapter 7 provide optimized n-variable, m-resilient functions with nonlinearities higher than all previous works [105, 33, 86].

The maximum nonlinearity obtained in [19] for 1-resilient functions was $2^{n-1} - 2^{\frac{n-1}{2}}$ for n odd and $2^{n-1} - 2^{\frac{n}{2}}$ for n even, where the degrees of the functions were strictly less than $n-2$. Our method provides same nonlinearity with optimized algebraic degree $n-2$. Moreover, for even $n \geq 16$, it is possible to get 1-resilient optimized functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n}{2}}$. This shows that it is possible to obtain higher nonlinearities for optimized functions than the upper bound derived in [19] for a certain subset of resilient functions.

4.4 Propagation Characteristics and Strict Avalanche Criteria

In this section we provide important results on propagation characteristics and strict avalanche criteria. We first present a result on nonlinearity of PC(1) functions.

Theorem 4.4.1 *For odd $n \geq 15$, it is possible to construct n-variable, $PC(1)$ functions with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^{\frac{n-15}{2}}$.*

Proof: Let $\overline{X} = (X_n, \ldots, X_1)$ and $\overline{\alpha} \in \{0,1\}^n$. Let $S_f = \{\overline{\alpha} \mid f(\overline{X}) \oplus f(\overline{X} \oplus \overline{\alpha})$ is balanced$\}$. We have found 15-variable functions f provided by Patterson and Wiedemann [89] such that there are 15 linearly independent vectors in S_f. Consider B_f to be a 15×15 nonsingular matrix whose rows are these 15 linearly independent vectors. Define $g(\overline{X}) = f(\overline{X}B_f)$. Then g has the same nonlinearity as f and satisfies PC(1).

For odd $n > 15$ we proceed inductively. Let g be an $(n-2)$-variable, PC(1) function with nonlinearity $2^{n-3} - 2^{\frac{n-3}{2}} + 20 \times 2^{\frac{n-17}{2}}$. Let h be a bent function on 2 variables which has the output column of the form 0001 and define $G = h \oplus g$. Thus G can also be seen as the concatenation $gggg^c$. Now take any n length binary vector $\overline{\alpha} = (\alpha_n, \alpha_{n-1}, \ldots, \alpha_1)$ with $wt(\overline{\alpha}) = 1$. Also we have $\overline{X} = (X_n, X_{n-1}, \ldots, X_1)$, and $\overline{X}' = (X_{n-2}, \ldots, X_1)$. Now we consider the following three cases.

Case 1. If $\overline{\alpha} = (\alpha_n = 0, \alpha_{n-1} = 0, \ldots, \alpha_1)$, then consider $\overline{\alpha}' = (\alpha_{n-2}, \ldots, \alpha_1)$. Note that $wt(\overline{\alpha}') = 1$. Thus, $wt(G(\overline{X}) \oplus G(\overline{X} \oplus \overline{\alpha})) = wt(g(\overline{X}') \oplus g(\overline{X}' \oplus \overline{\alpha}')) + wt(g(\overline{X}') \oplus g(\overline{X}' \oplus \overline{\alpha}')) + wt(g(\overline{X}') \oplus g(\overline{X}' \oplus \overline{\alpha}')) + wt(g^c(\overline{X}') \oplus g^c(\overline{X}' \oplus \overline{\alpha}')) = 4 \times 2^{n-3}$ (from induction hypothesis as g is PC(1)) $= 2^{n-1}$.

Case 2. If $\overline{\alpha} = (\alpha_n = 0, \alpha_{n-1} = 1, \ldots, \alpha_1 = 0)$, then $wt(G(\overline{X}) \oplus G(\overline{X} \oplus \overline{\alpha})) = wt(g \oplus g) + wt(g \oplus g) + wt(g \oplus g^c) + wt(g^c \oplus g) = 2^{n-1}$.

Case 3. If $\overline{\alpha} = (\alpha_n = 1, \alpha_{n-1} = 0, \ldots, \alpha_1 = 0)$, then $wt(G(\overline{X}) \oplus G(\overline{X} \oplus \overline{\alpha})) = wt(g \oplus g) + wt(g \oplus g^c) + wt(g \oplus g) + wt(g^c \oplus g) = 2^{n-1}$. Hence, G is PC(1) and since $G = h \oplus g$, we get $nl(G) = 2^{n-1} - 2^{\frac{n-1}{2}} + 20 \times 2^{\frac{n-15}{2}}$. ∎

The maximum possible nonlinearity of Boolean functions is equal to the covering radius of first order Reed-Muller codes. Patterson and Wiedemann showed that for odd $n \geq 15$ the covering radius and hence the maximum possible nonlinearity of an n-variable function exceeds $2^{n-1} - 2^{\frac{n-1}{2}}$. Seberry et al [104] showed that for odd $n \geq 29$, it is possible to construct balanced functions with nonlinearity greater than $2^{n-1} - 2^{\frac{n-1}{2}}$. Theorem 4.4.1 establishes a similar result for PC(1) functions of odd number of variables n for $n \geq 15$.

The following is a general construction of Boolean functions introduced in [58].

$$f(X_1, \ldots, X_s, Y_1, \ldots, Y_t) = [X_1, \ldots, X_s]Q[Y_1, \ldots, Y_t]^T \oplus g(X_1, \ldots, X_s), \qquad (*)$$

where Q is an $s \times t$ binary matrix and $g(X_1, \ldots, X_s)$ is any function.

Under certain conditions on Q, the function f satisfies PC(l) of order k (see [58]). Moreover, according to the proof of [58, Theorem 16], $nl(f) = 2^t nl(g)$ and $deg(f) = deg(g)$. It is possible to significantly improve the results of [58] by using functions constructed by the methods of Section 4.2.

Theorem 4.4.2 *For odd s, it is possible to construct $PC(l)$ of order k function f such that*

(a) $deg(f) = s - 1$ *and* $nl(f) \geq 2^{t+s-1} - 2^{t+\frac{s-1}{2}}$ *for* $3 \leq s \leq 13$,
(b) $deg(f) = s$ *and* $nl(f) > 2^{t+s-1} - 2^{t+\frac{s-1}{2}}$ *for* $s \geq 15$.

Proof : For $3 \leq s \leq 13$, s odd, we can consider $g \in \Omega_s$ as the function available from Theorem 4.2.3 with algebraic degree $s-1$ and nonlinearity $2^{s-1} - 2^{\frac{s-1}{2}}$. For $s \geq 15$, one can consider $g \in \Omega_s$ with nonlinearity $2^{s-1} - 2^{\frac{s-1}{2}} + 20 \times 2^{\frac{s-15}{2}} - 1$ and algebraic degree s. This can be obtained by considering a function on s variables with maximum known nonlinearity and then making $wt(g)$ odd by toggling one bit. This will provide the full algebraic degree and decrease the nonlinearity by at most 1 only. ■

For odd s, the corresponding result in [58] is $deg(f) = \frac{s-1}{2}$ and $nl(f) \geq 2^{t+s-1} - 2^{t+\frac{s-1}{2}}$ which is clearly improved in Theorem 4.4.2.

Now we show how to obtain maximum algebraic degree in this construction at the cost of small fall in nonlinearity. For odd s between 3 and 13, $deg(g)$ can be made s by changing one bit of g. This decreases $nl(g)$ by one. The corresponding parameters of f are $deg(f) = s$ and $nl(f) \geq 2^{t+s-1} - 2^{t+\frac{s-1}{2}} - 2^t$. For even s, the result in [58] is $deg(f) = \frac{s}{2}$ and $nl(f) \geq 2^{t+s-1} - 2^{t+\frac{s}{2}-1}$. As before by changing one bit of g we can ensure $deg(f) = s$ and $nl(f) \geq 2^{t+s-1} - 2^{t+\frac{s}{2}-1} - 2^t$.

Next we turn to the study of SAC(k) combined with the properties of balancedness, degree and nonlinearity. *This is the first time that all these properties are being considered together with SAC(k).* The proofs for the next few results are quite involved. Hence we present the constructions clearly and only sketch the proofs.

In [58], (*) has been used for the construction of SAC(k) function by setting $s = n-k-1$, $t = k+1$ and Q to be the $(n-k-1) \times (k+1)$ matrix whose all elements are 1. Under these conditions the function f takes the form $f(X_1, \ldots, X_n) = (X_1 \oplus \ldots \oplus X_{n-k-1})(X_{n-k} \oplus \ldots \oplus X_n) \oplus g(X_1, \ldots, X_{n-k-1})$. Moreover, it was shown that f is balanced if $| \{\overline{X} \mid g(\overline{X}) = 0, \overline{X}Q = 0\} | = | \{\overline{X} \mid g(\overline{X}) = 1, \overline{X}Q = 0\} |$ where $\overline{X} = (X_1, \ldots, X_{n-k-1})$. It is important to interpret this idea with respect to the truth table of g. This means that f is balanced if $\#\{\overline{X} \mid g(\overline{X}) = 0, wt(\overline{X}) = \text{ even }\} = \#\{\overline{X} \mid g(\overline{X}) = 1, wt(\overline{X}) = \text{ even }\}$. Thus, in the truth table we have to check for balancedness of g restricted to the rows where the weight of the input string is even. In half of such places g must be 0 and in the other half g must be 1. Motivated by this discussion we make the following definition of *brEven* (restricted balancedness with respect to inputs with even weight) and *brOdd* (restricted balancedness with respect to inputs with odd weight).

Definition 4.4.1 *Let* $g \in \Omega_p$ *and* $\overline{X} = (X_1, \ldots, X_p)$. *Then* g *is called brEven (resp. brOdd) if* $\#\{g(\overline{X}) = 0 \mid wt(\overline{X}) = \text{ even }\} = \#\{g(\overline{X}) = 1 \mid wt(\overline{X}) = \text{ even }\} = 2^{p-2}$
(resp. $\#\{g(\overline{X}) = 0 \mid wt(\overline{X}) = \text{ odd }\} = \#\{g(\overline{X}) = 1 \mid wt(\overline{X}) = \text{ odd }\} = 2^{p-2}$*).*

The next result is important as it shows that certain types of bent functions can be brEven. This allows us to obtain balanced SAC(k) functions with very high nonlinearity which could not be obtained in [58].

Proposition 4.4.1 *For p even, it is possible to construct bent functions $g \in \Omega_p$ which are brEven.*

Proof : First note that g is brEven iff g^c is brEven. Let $q = 2^{\frac{p}{2}}$. For $0 \leq i \leq q-1$ let $l_i \in L(\frac{p}{2})$ be the linear function $a_{\frac{p}{2}} X_{\frac{p}{2}} \oplus \ldots \oplus a_1 X_1$, where $a_{\frac{p}{2}} \ldots a_1$ is the $\frac{p}{2}$-bit binary expansion of i. We provide construction of bent functions $g(X_1, \ldots, X_p)$ which are brEven. Let $\overline{X} = (X_1, \ldots, X_p)$.
Case 1:
$\frac{p}{2} \equiv 1 \bmod 2$. Let $g = l_0 f_1 \ldots f_{q-2} l_{q-1}$, where $f_1, \ldots, f_{q-2} \in \{l_1, \ldots, l_{q-2}, l_1^c, \ldots, l_{q-2}^c\}$ and for $i \neq j$, $f_i \neq f_j$ and $f_i \neq f_j^c$. It is well known that such a g is bent [95]. We show that g is brEven. First we have the following three results which we state without proofs.
(a) $\#\{l_0(X_1, \ldots, X_{\frac{p}{2}}) = 0 \mid wt(X_1, \ldots, X_{\frac{p}{2}}) = \text{ even } \} = 2^{\frac{p}{2}-1}$ and
$\#\{l_0(X_1, \ldots, X_{\frac{p}{2}}) = 1 \mid wt(X_1, \ldots, X_{\frac{p}{2}}) = \text{ even } \} = 0$.
(b) Since the f_i's are degenerate affine functions in $L(\frac{p}{2})$, it is possible to show that individually they are both brEven and brOdd.
(c) Using the fact that $q = \frac{p}{2}$ is odd and $l_{q-1} = X_1 \oplus \ldots \oplus X_{\frac{p}{2}}$, it is possible to show,
$\#\{l_{q-1}(X_1, \ldots, X_{\frac{p}{2}}) = 0 \mid wt(X_1, \ldots, X_{\frac{p}{2}}) = \text{ even } \} = 0$ and
$\#\{l_{q-1}(X_1, \ldots, X_{\frac{p}{2}}) = 1 \mid wt(X_1, \ldots, X_{\frac{p}{2}}) = \text{ even } \} = 2^{\frac{p}{2}-1}$.
Then using $wt(X_1, \ldots, X_p) = wt(X_1, \ldots, X_{\frac{p}{2}}) + wt(X_{\frac{p}{2}+1}, \ldots, X_p)$ and the fact that g is concatenation of $l_0, f_1, \ldots, f_{q-2}, l_{q-1}$ it is possible to show that g is brEven.
Case 2:
For $\frac{p}{2} \equiv 0 \bmod 2$, the result is true for bent functions of the form $g = l_0^c f_1 \ldots f_{q-2} l_{q-1}$. ■

In [58, Theorem 32] it has been stated that for $n-k-1 =$ even, there exists balanced SAC(k) functions such that $deg(f) = n-k-2$. The question whether such functions with algebraic degree $n-k-1$ exists has been left as an open question. The next result shows the existence of such functions which proves that the bound on algebraic degree provided in [93] is indeed tight for $k \leq \frac{n}{2} - 1$.

Theorem 4.4.3 *Let $(n-k-1) \geq (k+1)$, i.e. $k \leq \frac{n}{2} - 1$ and $n-k-1 =$ even. Then it is possible to construct balanced SAC(k) function $f \in \Omega_n$ such that $deg(f) = n-k-1$. Moreover $nl(f) = 2^{n-1} - 2^{\frac{n+k-1}{2}} - 2^{k+1}$.*

Proof : Use a bent function $g \in \Omega_{n-k-1}$ which is brEven. Out of the 2^{n-k-1} bit positions in g (in the output column of the truth table), there are 2^{n-k-2} positions where $wt(X_1, \ldots, X_{n-k-1}) =$ odd and the value of g at these positions can be toggled without

disturbing the brEven property. Since g is bent, $wt(g) =$ even. Thus we choose a row j in the truth table where $wt(X_1, \ldots, X_{n-k-1}) =$ odd and construct g' by toggling the output bit. Thus $wt(g') = wt(g) \pm 1 =$ odd. Hence by Proposition 4.1.1, $deg(g') = n-k-1$. Thus, $f(X_1, \ldots, X_n) = (X_1 \oplus \ldots \oplus X_{n-k-1})(X_{n-k} \oplus \ldots \oplus X_n) \oplus g'(X_1, \ldots, X_{n-k-1})$ is balanced SAC(k) with algebraic degree $n-k-1$. Also $nl(g') = nl(g) - 1 = 2^{n-k-2} - 2^{\frac{n-k-1}{2}-1} - 1$. Now, $nl(f) = 2^{k+1} \times nl(g') = 2^{n-1} - 2^{\frac{n+k-1}{2}} - 2^{k+1}$. ■

Example 4.4.1 *We provide this example for illustration of Proposition 4.4.1 and Theorem 4.4.3.*

Let $n-k-1 = m = 4$. From Proposition 4.4.1, we take g as $\underline{1}111\underline{0}1\underline{0}10\underline{0}11\underline{0}110$, the output column of the truth table. The underlined values correspond to the input variables with even weight. Note that $\#\{g(X_1, \ldots, X_m) = 1 \mid wt(X_1, \ldots, X_m) =$ even $\} = \#\{g(X_1, \ldots, X_m) = 0 \mid wt(X_1, \ldots, X_m) =$ even $\}$. Thus g is brEven. Also $wt(g) = 10$. Now we construct $g' = \underline{1}011\underline{0}1\underline{0}10\underline{0}11\underline{0}110$, by changing the 2nd bit of g from 1 to 0. Note that g' is also brEven and $wt(g') = 9 =$ odd, which means $deg(g') = 4$.

Next we provide similar results for odd $n-k-1$. The result is extremely important in the sense that the functions constructed in [88, 89] can be modified to get restricted balancedness and hence can be used in the construction of highly nonlinear, balanced SAC(k) functions. *We know of no other place where the functions provided by Patterson and Wiedemann [88, 89] have been used in the construction of SAC(k) functions.*

Proposition 4.4.2 *For p odd, it is possible to construct $g \in \Omega_p$ with nonlinearity (i) $2^{p-1} - 2^{\frac{p-1}{2}}$ for $p \leq 13$ and (ii) $2^{p-1} - 2^{\frac{p-1}{2}} + 20 \times 2^{\frac{p-15}{2}}$ for $p \geq 15$ which is brEven.*

Proof : First note that if any function $f(X_1, \ldots, X_n)$ is brEven, then $f(X_1 \oplus \alpha_1, \ldots, X_n \oplus \alpha_n)$ is brOdd when $wt(\alpha_1, \ldots, \alpha_n)$ is odd and vice versa. Also $nl(f(X_1, \ldots, X_n)) = nl(f(X_1 \oplus \alpha_1, \ldots, X_n \oplus \alpha_n))$.

For odd $p \leq 13$, choose $f_3 \in \Omega_{p-1}$, a brEven bent function, as given in Proposition 4.4.1. Then construct $f_2 \in \Omega_{p-1}$, such that, $f_2 = f_3(X_1 \oplus \alpha_1, \ldots, X_{p-1} \oplus \alpha_{p-1})$ where $wt(\alpha_1, \ldots, \alpha_{p-1})$ is odd. Thus, f_2 is a brOdd function. Now construct a function $F \in \Omega_p$, where F is concatenation of f_3 and f_2, i.e. $F = (1 \oplus X_p)f_3 \oplus X_p f_2$. Then F is brEven and $nl(F) = 2^{p-1} - 2^{\frac{p-1}{2}}$.

For $p \geq 15$ the construction is different. Let $f_1 \in \Omega_{15}$ be one of the functions constructed in [89]. Note that $nl(f_1) = 2^{14} - 2^7 + 20$. Now consider the 32768 functions of the form $f_1 \oplus l$, where $l \in L(15)$. We have found functions among these which are brOdd (but none of which are brEven). Let $f_2(X_1, \ldots, X_{15})$ be such a brOdd function. Then $f_3(X_1, \ldots, X_{15}) = f_2(X_1 \oplus$

$\alpha_1, \ldots, X_{15} \oplus \alpha_{15})$ is brEven when $wt(\alpha_1, \ldots, \alpha_{15})$ is odd. Note that $nl(f_2) = nl(f_3) = nl(f_1)$. This settles the base case.

Now we consider that for all odd p, $15 \leq p \leq m$, (m odd), there exists a brEven function $f \in \Omega_p$ with $nl(f) = 2^{p-1} - 2^{\frac{p-1}{2}} + 20 \times 2^{\frac{p-15}{2}}$. Let us take $f_3 \in \Omega_m$ be a brEven function with $nl(f_3) = 2^{m-1} - 2^{\frac{m-1}{2}} + 20 \times 2^{\frac{m-15}{2}}$. Then we can construct brOdd function $f_2(X_1, \ldots, X_m) = f_3(X_1 \oplus \alpha_1, \ldots, X_p \oplus \alpha_m)$ where $wt(\alpha_1, \ldots, \alpha_m)$ is odd. Construct $F \in \Omega_{m+2}$ where F can be seen as the concatenation $f_3 f_2 f_2 f_3^c$. Then F is also brEven. Now, any linear function on $(m+2)$ variables can be seen as any of the four functions $llll, ll^c ll^c, lll^c l^c, ll^c l^c l$, where $l \in L(m)$. Thus, considering the values of $d(f_3 f_2 f_2 f_3^c, llll), d(f_3 f_2 f_2 f_3^c, ll^c ll^c), d(f_3 f_2 f_2 f_3^c, lll^c l^c)$, $d(f_3 f_2 f_2 f_3^c, ll^c l^c l)$, we get, $nl(F) = 2^{m+2-1} - 2^{\frac{m+2-1}{2}} + 20 \times 2^{\frac{m+2-15}{2}}$. This proves the inductive step. ∎

The construction in the above theorem can also be seen in the following way. Consider a brOdd function f_2 and a brEven function f_3 on 15 variables with $nl(f_2) = nl(f_3) = 2^{14} - 2^7 + 20 \times 2^0$. Now we show the construction of a brEven function on $15 + 2k$ variables. Let $g(Y_1, \ldots, Y_{2k})$ be a bent function on $2k$ variables. Define $F \in \Omega_{15+2k}$ as follows. $F = (Y_1 \oplus \ldots \oplus Y_{2k})(g \oplus f_2) \oplus (1 \oplus Y_1 \oplus \ldots \oplus Y_{2k})(g \oplus f_3)$. Then F is brEven and $nl(F) = 2^{14+2k} - 2^{7+k} + 20 \times 2^k$.

Theorem 4.4.4 *Let $(n-k-1) \geq (k+1)$, i.e. $k \leq \frac{n}{2} - 1$ and $n-k-1 =$ odd. Then it is possible to construct balanced $SAC(k)$ function $f \in \Omega_n$ such that $deg(f) = n-k-1$. Moreover, for $3 \leq n-k-1 \leq 13$, $nl(f) = 2^{n-1} - 2^{\frac{n+k}{2}} - 2^{k+1}$ and for $n-k-1 \geq 15$, $nl(f) = 2^{n-1} - 2^{\frac{n+k}{2}} + 20 \times 2^{\frac{n+k-14}{2}} - 2^{k+1}$.*

Proof : We start with the functions available in Proposition 4.4.2. Then the proof is similar to the proof of Theorem 4.4.3. ∎

This shows that it is possible to construct highly nonlinear balanced functions satisfying SAC(k) with maximum possible algebraic degree $n-k-1$. Functions with all these criteria at the same time has not been considered earlier.

Now we present an interesting result combining resiliency and propagation characteristics. In [105, Theorem 15], propagation criterion of m-resilient functions has been studied. Those functions satisfy propagation criteria with a specific set of vectors. However, they do not satisfy even PC(1) as propagation criteria is not satisfied for some vectors of weight 1. For n even, we present a construction to provide resilient functions in Ω_n which satisfy PC($\frac{n}{2} - 1$).

Theorem 4.4.5 *It is possible to construct 1-resilient functions in Ω_n, n even, with nonlinearity $2^{n-1} - 2^{\frac{n}{2}}$ and algebraic degree $\frac{n}{2} - 1$ which satisfy $PC(\frac{n}{2} - 1)$.*

Proof : Let $f \in \Omega_{n-2}$ be a bent function, n even. Then $F(X_1, \ldots, X_{n-1}) = (1 \oplus X_{n-1}) f(X_1, \ldots, X_{n-2}) \oplus X_{n-1}(1 \oplus f(X_1 \oplus \alpha_1, \ldots, X_{n-2} \oplus \alpha_{n-2}))$ is balanced and satis-

fies propagation criterion with respect to all nonzero vectors except $(\alpha_1, \ldots, \alpha_{n-2}, 1)$. Also $nl(F) = 2^{n-2} - 2^{\frac{n-2}{2}}$.

Let $G(X_1, \ldots, X_n) = (1 \oplus X_n)F(X_1, \ldots, X_{n-1}) \oplus X_n(F(X_1 \oplus \beta_1, \ldots, X_{n-1} \oplus \beta_{n-1}))$. Then G is balanced and satisfies propagation criterion with respect to all nonzero vectors except $\overline{\alpha} = (\alpha_1, \ldots, \alpha_{n-2}, \alpha_{n-1} = 1, \alpha_n = 0)$, $\overline{\beta} = (\beta_1, \ldots, \beta_{n-1}, \beta_n = 1)$ and $\overline{\alpha} \oplus \overline{\beta}$. Also G is balanced and $nl(G) = 2^{n-1} - 2^{\frac{n}{2}}$.

Take $(\alpha_1, \alpha_2, \ldots, \alpha_{n-2})$ in the construction of F in Ω_{n-1} from $f \in \Omega_{n-2}$ so that $wt(\alpha_1, \alpha_2, \ldots, \alpha_{n-2}) = \frac{n}{2} - 1$. Also $G(X_1, \ldots, X_n) = (1 \oplus X_n)F(X_1, \ldots, X_{n-1}) \oplus X_n(F(X_1 \oplus 1, \ldots, X_{n-1} \oplus 1)$ is correlation immune [12]. Since F is balanced, G is also balanced which proves that G is 1-resilient. Now consider $\overline{\alpha} = (\alpha_1, \ldots, \alpha_{n-2}, \alpha_{n-1} = 1, \alpha_n = 0)$, $\overline{\beta} = (\beta_1 = 1, \ldots, \beta_{n+1} = 1, \beta_{n+2} = 1)$. Since $wt(\overline{\alpha}) = \frac{n}{2} - 1 + 1$ and $wt(\overline{\beta}) = n$ we get, $wt(\overline{\alpha} \oplus \overline{\beta}) = \frac{n}{2}$. Note that G satisfies propagation criterion with respect to all the nonzero vectors except $\overline{\alpha}, \overline{\beta}, \overline{\alpha} \oplus \overline{\beta}$ and hence G satisfies PC$(\frac{n}{2} - 1)$.

Since $f \in \Omega_{n-2}$ is bent, it is possible to construct f with algebraic degree $\frac{n}{2} - 1$ and $deg(G) = deg(f)$. ■

Chapter 5

Hamming Weights of Correlation Immune Boolean Functions

The set of correlation immune (CI) Boolean functions can be partitioned into several disjoint subsets depending on the Hamming weight of their output column. We show that the number of n variable CI functions of Hamming weight $2a+2$ is strictly greater than the number of such functions of weight $2a$ for $2a < 2^{n-1}$. This seemingly intuitive result turns out to be quite difficult to be proved. The combinatorial structure of CI functions revealed here reduces the enumeration problem of CI functions to the enumeration problem of balanced CI functions. Moreover, we characterize the CI functions considering the Hamming distance between the function (interpreted as a binary string of length 2^n) and its reverse binary string. Our results provide a different direction in comparison with the existing results for enumeration of correlation immune functions.

5.1 Introduction

The concept of correlation immune Boolean functions was introduced by Siegenthaler [109]. Recently the enumeration of correlation immune Boolean functions has received a lot of attention as evident from [80, 118, 81, 65]. The set of n-variable Boolean functions can be partitioned into $2^n + 1$ disjoint sets depending on the Hamming weights of their output columns. In this context it is natural to consider the set of CI functions restricted to a particular weight. One particularly interesting question that immediately arises is how does the number of CI functions of a certain weight compare to the number of CI functions of a greater weight. We completely settle this question by showing that (a) the number of CI functions of odd weight is 0, (b) the number of CI functions of weight $2a$ is equal to number of CI functions of weight $2^n - 2a$ and (c) the number of CI functions of weight $2a$ is strictly

lesser than the number of CI functions of weight $2a+2$ for $2a < 2^{n-1}$. Parts (a) and (b) are easy and though (c) is intuitive proving it is a nontrivial task. Our proof technique throws new light on the inherent combinatorial nature of CI functions and also reduces the enumeration problem for CI functions to the enumeration problem for balanced CI functions. Also we characterize the correlation immune functions using the Hamming distance of a CI function with its reverse. Our results explain the hierarchical structure on the number of CI Boolean functions.

We interpret a Boolean function f as a binary string of length 2^n, given by the output column in the truth table and $wt(f)$ means the number of 1's (Hamming weight) in the string f. By f^u, we denote the upper half of f, whereas f^l denotes the lower half (each half is of length 2^{n-1}). The string f^r is the reverse of string f and f^c is the bitwise complement of f. By $S[\tau]$ we mean the τth bit in the binary string S. Also, $\#(\phi)$ counts the number of outcomes favourable to the event ϕ. The notation $(A \mid B)$ denotes the outcomes favorable to A given that B has already occurred. By $d(S_1, S_2)$ we denote the Hamming distance between two strings S_1, S_2 of same length (say λ). Also, the number of places in which S_1, S_2 matches is denoted by $M(S_1, S_2)$, i.e., $M(S_1, S_2) = \lambda - d(S_1, S_2)$. Let $M_0(f_1, f_2) = \#(f_1[i] = f_2[i] = 0)$ and $M_1(f_1, f_2) = \#(f_1[i] = f_2[i] = 1)$, $0 \leq i \leq 2^n - 1$. Thus, $M_0(f_1, f_2) + M_1(f_1, f_2) = M(f_1, f_2)$.

Next we define correlation immunity of a Boolean function [109, 80].

Definition 5.1.1 *Let f be a Boolean function of n input variables $\{X_1, X_2, \ldots, X_n\}$. Then f is correlation immune if $Prob(f = X_i) = \frac{1}{2}, \forall i,\ 1 \leq i \leq n$.*

The set of all Boolean functions of n variables is denoted by Ω_n, and the set of all correlation immune Boolean functions of n variables is denoted by A_n. Further, $CIW_n(a) = \{f \in A_n \mid wt(f) = a\}$ denotes all n-variable CI functions of weight a and $C_n(a) = \mid CIW_n(a) \mid$ denotes the number of such functions.

We here show that, (a) $C_n(2a+1) = 0$, (b) $C_n(2a) = C_n(2^n - 2a)$ and (c) $C_n(2a) < C_n(2a+2)$ for $2a < 2^{n-1}$. The following simple result settles (a).

Proposition 5.1.1 *$Prob(f = X_i) = \frac{1}{2}$ iff $\#(f = 1 \mid X_i = 0) = \#(f = 1 \mid X_i = 1)$ $\forall i,\ 1 \leq i \leq n$. Consequently, $C_n(2a+1) = 0$, for $a \geq 0$.*

5.2 Weight Distribution

It is easy to see that for $a < 2^{n-1}$, the number of n-variable functions of weight a is less than the number of n-variable functions of weight $a+1$. This follows from simple properties of binomial coefficients. It is then intuitive to expect same kind of results for correlation

immune functions as well. In this section, we prove such a result. First we show the following which is analogous to the identity $\binom{m}{a} = \binom{m}{m-a}$.

Proposition 5.2.1 $C_n(2a) = C_n(2^n - 2a)$.

Proof : The result follows on noting that $f \in A_n$ iff $f^c \in A_n$. ■

Based on Proposition 5.2.1, in the rest of this section we will consider $2a < 2^{n-1}$, *i.e.* $2a \leq 2^{n-1} - 2$ *unless otherwise mentioned.*

Proposition 5.2.2 *Let* $f \in CIW_n(2a)$, $n \geq 2$. *Then* $M(f, f^r) \equiv 0 \bmod 4$. *Consequently,* $d(f, f^r)$ *is also congruent to* $0 \bmod 4$.

Proof : Let f^u, f^l be the top and bottom halves (of equal length) of f respectively. Since, $f \in A_n$, we have $wt(f^u) = wt(f^l) = a$. Let there be k places out of the a 1's in f^u part where the corresponding positions in $(f^l)^r$ do not match, i.e., $M_1(f^u, (f^l)^r) = (a-k)$. Thus, there are k places out of $(2^{n-1} - a)$ 0's in $(f^l)^r$ part where the corresponding positions in f^u do not match, which gives $M_0(f^u, (f^l)^r) = 2^{n-1} - a - k$. Hence, $M(f, f^r) = 2M(f^u, (f^l)^r) = 2((a-k) + (2^{n-1} - a - k)) = 2^n - 4k \equiv 0 \bmod 4$. ■

From the argument of the proof of Proposition 5.2.2, we get the following result.

Proposition 5.2.3 *Let* $f \in CIW_n(2a)$ *and* $M(f, f^r) = x$. *Then* $M_0(f, f^r) = 2^{n-1} - 2a + \frac{x}{2}$ *and* $M_1(f, f^r) = -2^{n-1} + 2a + \frac{x}{2}$. *Consequently,* $M_0(f, f^r) - M_1(f, f^r) = 2^n - 4a$.

Now we provide a construction technique of $g \in CIW_n(2a+2)$ from $f \in CIW_n(2a)$ and vice versa.

Definition 5.2.1 *Let* $f, g \in \Omega_n$ *and there exists* i_0, i_1 *with* $i_0 + i_1 = 2^n - 1$, *such that (1)* $f[i_0] = f[i_1] = 0$, *(2)* $g[i_0] = g[i_1] = 1$ *and (3)* $f[j] = g[j]$ *if* $j \neq i_0, i_1$.
Then we say that f, g *are palindromically related.*

Note that values of just a specific pair of positions are toggled and the positions are at the same distances from top and bottom of the function string. It is important to note that two functions f, g are palindromically related means that $d(f, g) = 2$, i.e., $M(f, g) = 2^n - 2$.

Proposition 5.2.4 *If* f, g *be palindromically related then* $M(f, f^r) = M(g, g^r)$. *Equivalently,* $d(f, f^r) = d(g, g^r)$.

The following result shows the importance of Definition 5.2.1.

Theorem 5.2.1 *Let* f, g *be palindromically related. Then* $f \in A_n$ *iff* $g \in A_n$.

Proof : Since $f \in CIW_n(2a)$, we have $\#(f = 1 \mid X_i = 0) = \#(f = \mathbf{1} \mid X_i = 1) = a$ for all i. Also there exists τ such that, $f[\tau] = f[2^n - 1 - \tau] = 0$. Consider the column of X_i in the truth table as a binary string. Note that, $X_i[\tau] = (X_i[2^n - 1 - \tau])^c$. Thus, if we consider the function g, then we have, $\#(g = 1 \mid X_i = 0) = \#(g = 1 \mid X_i = 1) = a + 1$. Thus $g \in A_n$. The other direction can be proved similarly. ■

Corollary 5.2.1 *All palindromic functions are CI.*

Proof : The identity function 0 is trivially correlation immune. The result then follows from Theorem 5.2.1 by induction on the weight of a palindrome. ■

This result has also been proved differently [80]. As an immediate consequence of Corollary 5.2.1 we have $C_n(2a) \geq \binom{2^{n-1}}{a}$. Interestingly, the exact proportion of $C_n(2a)$ among all functions of weight $2a$ is an open question. However, we have the exact result for $2a = 2$. Since, the set $CIW_n(2)$ contains only the palindromes, we get $C_n(2) = 2^{n-1}$.

If we take $f \in CIW_n(2a)$, $2a < 2^{n-1}$, then from Proposition 5.2.3, we have $M_0(f, f^r) > 0$. Thus there exists at least one position τ such that $f[\tau] = f[2^n - 1 - \tau] = 0$. Then using Definition 5.2.1, we can get some $g \in CIW_n(2a + 2)$, by replacing the pair of 0's by a pair of 1's. Moreover, if there exists more than one τ, such that $f[\tau] = f[2^n - 1 - \tau] = 0$, then different functions of $CIW_n(2a + 2)$ can be constructed from f. Let us now consider the other way around. Let $g \in CIW_n(2a + 2)$. If $M_1(g, g^r) > 0$, then using the Definition 5.2.1, some $f \in CIW_n(2a)$ can be found. However, it is important to note that, there may exist some $g \in CIW_n(2a + 2)$ with $M_1(g, g^r) = 0$. In that case it is not possible to get a function $f \in CIW_n(2a)$ by changing one pair of positions. Motivated by this discussion we make the following definition.

Definition 5.2.2 *Let G_n be an undirected graph where the vertices are the elements of A_n and two vertices are connected if they are palindromically related. We call such a graph an n-variable correlation immune graph.*

The following theorem provides an important property of G_n.

Theorem 5.2.2 *If two vertices f, g of the CI graph G_n belong to the same component of G_n then $M(f, f^r) = M(g, g^r)$.*

Proof : The path $u = u_0, u_1, \ldots, u_{k-1}, u_k = v$ exists in G_n iff u_i, u_{i+1}, $0 \leq i \leq k - 1$, are palindromically related. The result then follows from Proposition 5.2.4. ■

Next we define another graph which is basically a subgraph of the CI graph.

Definition 5.2.3 *By $G_n(2a, 2a + 2)$, we define an undirected bipartite graph such that $G_n(2a, 2a + 2) = (CIW_n(2a) \cup CIW_n(2a + 2), E)$, where there is an edge between $f \in CIW_n(2a)$ and $g \in CIW_n(2a + 2)$ if they are palindromically related.*

The following defines a special type of bipartite graph.

Definition 5.2.4 *Let $G = (V_1 \cup V_2, E)$ be a connected bipartite graph. Then G is called homogeneous if all the vertices of V_1 are of same degree d_1 and all the vertices of V_2 are of same degree d_2.*

Homogeneous bipartite graphs have the following simple property which will prove to be useful later.

Proposition 5.2.5 *Let $G = (V_1 \cup V_2, E)$ be homogeneous graph. Let degree of each vertex of V_1 be d_1 and degree of each vertex of V_2 be d_2. Then $\mid V_1 \mid \times d_1 = \mid V_2 \mid \times d_2 = \mid E \mid$. Consequently, if $d_1 > d_2$ then $\mid V_1 \mid < \mid V_2 \mid$.*

Our next task is to identify the disjoint bipartite subgraphs of $G_n(2a, 2a+2)$ which are homogeneous. We need the following additional notations. Let $CIW_{n,x}(2a) = \{f \in CIW_n(2a) \mid M(f, f^r) = x\}$ and $C_{n,x}(2a) = \mid CIW_{n,x}(2a) \mid$. By $G_{n,x}(2a, 2a+2)$ we mean the subgraph of $G_n(2a, 2a+2)$ induced by the vertices of $CIW_{n,x}(2a) \cup CIW_{n,x}(2a+2)$. The following relates $G_{n,x}(2a, 2a+2)$ to $G_n(2a, 2a+2)$.

Lemma 5.2.1 *The subgraphs $G_{n,x}(2a, 2a+2)$ of the graph $G_n(2a, 2a+2)$ are homogeneous for all possible values of x.*

Proof : This follows from Theorem 5.2.2 and Proposition 5.2.4. First consider $f, f_1 \in CIW_{n,x}(2a)$. Then, $M(f, f^r) = M(f_1, f_1^r) = x$. Hence, from Proposition 5.2.3 degree of f, $degree(f) = \frac{M_0(f,f^r)}{2} = \frac{M_0(f_1,f_1^r)}{2} = degree(f_1)$. Similarly, for $g, g_1 \in CIW_{n,x}(2a+2)$, $degree(g) = \frac{M_1(g,g^r)}{2} = \frac{M_1(g_1,g_1^r)}{2} = degree(g_1)$. Thus $G_{n,x}(2a, 2a+2)$ is homogeneous. ■

Lemma 5.2.2 *Let f, g be vertices of $G_{n,x}(2a, 2a+2)$ where, $f \in CIW_{n,x}(2a)$, $g \in CIW_{n,x}(2a+2)$ and $2a < 2^{n-1}$. Then, the degree of f is $degree(f) = \frac{M_0(f,f^r)}{2}$ and the degree of g is $degree(g) = \frac{M_1(g,g^r)}{2}$ with $degree(f) > degree(g) > 0$. Consequently, $C_{n,x}(2a) < C_{n,x}(2a+2)$.*

Proof : Note that, $degree(f) = \frac{1}{2}M_0(f, f^r)$ and $degree(g) = \frac{1}{2}M_1(g, g^r)$. Now we use Proposition 5.2.3. We have, $M_0(f, f^r) = 2^{n-1} - 2a + \frac{x}{2}$ and $M_1(g, g^r) = -2^{n-1} + 2a + 2 + \frac{x}{2}$. Consequently, $M_0(f, f^r) - M_1(g, g^r) = 2^n - 4a - 2$ which gives $M_0(f, f^r) > M_1(g, g^r)$ since $2a \leq 2^{n-1} - 2$. This gives, $degree(f) > degree(g)$. Note that, $M_1(f, f^r) = -2^{n-1} + 2a + \frac{x}{2} \geq 0$, thus, $M_1(g, g^r) = -2^{n-1} + 2a + 2 + \frac{x}{2} \geq 2$ and hence $degree(g) > 0$. The last statement follows from Proposition 5.2.5. ■

It is important to note that there may be more than one components in $G_{n,x}(2a, 2a+2)$. Since $G_{n,x}(2a, 2a+2)$ itself is homogeneous bipartite, the components are also of the same structure.

Theorem 5.2.3 *$C_n(2a) < C_n(2a+2)$ for $2a < 2^{n-1}$ and $C_n(2a) > C_n(2a+2)$ for $2a \geq 2^{n-1}$.*

Proof : Using Proposition 5.2.1 it is sufficient to show $C_n(2a) < C_n(2a+2)$ for $2a < 2^{n-1}$. Let there be t distinct values $x_1, x_2, \ldots, x_t$ of $M(f, f^r)$ for $f \in CIW_n(2a)$. From Lemma 5.2.2, we have, $C_{n,x_i}(2a) < C_{n,x_i}(2a+2)$. Also, $CIW_{n,x_i}(2a) \cap CIW_{n,x_j}(2a) = \emptyset$ for $x_i \neq x_j$. Hence,

$$C_n(2a) = \sum_{i=1}^{t} C_{n,x_i}(2a) < \sum_{i=1}^{t} C_{n,x_i}(2a+2) \leq C_n(2a+2).$$

Thus, $C_n(2a) < C_n(2a+2)$ for $2a < 2^{n-1}$. ■

5.3 Balanced Functions

In this section we show that if the set of balanced correlation immune function $CIW_n(2^{n-1})$ can be enumerated with the cardinality of each partition $CIW_{n,x}(2^{n-1})$ separately, then the exact enumeration of A_n is possible. First we express the proportional cardinality of two sets which follows from Proposition 5.2.3.

Lemma 5.3.1 $\frac{C_{n,x}(2^{n-1}-2(i+1))}{C_{n,x}(2^{n-1}-2i)} = \frac{\frac{x}{2}-2i}{\frac{x}{2}+2i+2}$, *for* $\frac{x}{2} - 2i > 0, i \geq 0$.

Let $MATCH_n = \{x \mid M(f, f^r) = x,$ for some $f \in CIW_n(2a)$, $2a \leq 2^{n-1}\}$ and $BMATCH_n = \{x \mid M(f, f^r) = x,$ for some $f \in CIW_n(2^{n-1})\}$. The next result shows $MATCH_n = BMATCH_n$.

Lemma 5.3.2 *If $f \in CIW_{n,x}(2a)$, $2a < 2^{n-1}$, then there exists a function g such that $g \in CIW_{n,x}(2^{n-1})$.*

Proof : Let $f \in CIW_{n,x}(2a)$. Then there is a path $f = f_0, f_1, \ldots, f_k = g$ of length $k = 2^{n-2} - a$ in the CI graph G_n, where (a) $wt(f_i) = 2 + wt(f_{i-1})$ and (b) $M_0(f_i, f_i^r) = 2^{n-1} - (2a + 2i) + \frac{x}{2}$ for $i \geq 1$. Thus $g \in CIW_n(2^{n-1})$. Using Proposition 5.2.4, we have $M(f, f^r) = M(g, g^r)$ and so $g \in CIW_{n,x}(2^{n-1})$. ■

Let $f \in A_n$ and $M(f, f^r) = x_j$. Now the previous result shows that there exists such an f with $wt(f) = 2^{n-1}$ (i.e. f balanced). Now what is the minimum weight of f such that $f \in A_n$ and $M(f, f^r) = x_j$. Since, $M(f, f^r) = x_j$, we can fill all the x_j places with 0's. Now, $d(f, f^r) = 2^n - x_j$ and the number of 1's in upper half f^u is equal to the number of 0's in lower half f^l. Thus, $wt(f) = 2^{n-1} - \frac{x_j}{2}$. On the other hand, if $wt(f) < 2^{n-1} - \frac{x_j}{2}$, then f can not be a CI function with $M(f, f^r) = x_j$. It is clear that the number of balanced CI functions f with $M(f, f^r) = x_j$ is larger than the number of such CI functions with other weights. Lemma 5.3.1 provides one such ratio. The next result provides another such ratio.

	$C_{4,0}(2a)$	$C_{4,8}(2a)$	$C_{4,16}(2a)$	$C_4(2a)$ = row total
2a = 0	0	0	1	1
2a = 2	0	0	8	8
2a = 4	0	24	28	52
2a = 6	0	96	56	152
2a = 8	8	144	70	222

Table 5.1: Enumerating A_4

Lemma 5.3.3 $C_{n,x_j}(2^{n-1} - \frac{x_j}{2}) = \frac{((\frac{x_j}{4})!)^2}{(\frac{x_j}{2})!} C_{n,x_j}(2^{n-1})$.

Proof : Using Lemma 5.3.1 we get, for $i > 0$, $C_{n,x_j}(2^{n-1} - 2i) = C_{n,x_j}(2^{n-1}) \prod_{k=0}^{i-1} \frac{\frac{x_j}{2} - 2k}{\frac{x_j}{2} + 2k + 2}$. Putting $2i = \frac{x_j}{2}$ the result follows. ■

Let $BMATCH_n = \{x_1, \ldots, x_t\}$. For $1 \leq j \leq t$, $0 < 2i \leq 2^{n-1}$, $\sigma_{j,2i} = 1$ if $\frac{x_j}{2} - 2i > 0$ and $\sigma_{j,2i} = 0$, otherwise.

Theorem 5.3.1 $C_n(2^{n-1} - 2i) = \sum_{j=1}^{t} \sigma_{j,2i} C_{n,x_j}(2^{n-1}) \prod_{k=0}^{i-1} \frac{\frac{x_j}{2} - 2k}{\frac{x_j}{2} + 2k + 2}$.

Proof : Using Lemma 5.3.1 we get, for $i > 0$, $C_{n,x_j}(2^{n-1} - 2i) = C_{n,x_j}(2^{n-1}) \prod_{k=0}^{i-1} \frac{\frac{x_j}{2} - 2k}{\frac{x_j}{2} + 2k + 2}$. Since (a) $CIW_n(2^{n-1} - 2i)$ is a disjoint union of the sets $CIW_{n,x_j}(2^{n-1} - 2i)$ for $x_j \in BMATCH_n$ and (b) $CIW_{n,x_j}(2^{n-1} - 2i) = \emptyset$ iff $\sigma_{j,2i} = 0$, the result holds. ■

Theorem 5.3.2 $\mid A_n \mid = C_n(2^{n-1}) + 2 \sum_{j=1}^{t} C_{n,x_j}(2^{n-1}) \sum_{i=1}^{\frac{x_j}{4}} \prod_{k=0}^{i-1} \frac{\frac{x_j}{2} - 2k}{\frac{x_j}{2} + 2k + 2}$.

Proof : This follows from Theorem 5.3.1 and Proposition 5.2.1. The expression $\sigma_{j,2i}$ is removed by summing i from 1 to $\frac{x_j}{4}$. ■

Table 5.1 provides an example which shows the result for 4 variable correlation immune Boolean functions. This gives $\mid A_4 \mid = 2 \times (1 + 8 + 52 + 152) + 222 = 648$.

5.4 Values of $d(f, f^r)$

In the Proposition 5.2.2 we have found that for any correlation immune function f, $d(f, f^r)$ is congruent to 0 mod 4. Now this is an interesting problem to identify exactly what values can $d(f, f^r)$ take. The following series of results are important in that direction.

Proposition 5.4.1 *For $n \geq 3$ and $f \in A_n$, $d(f, f^r) \neq 4$.*

Proof : Let, in contrary, $d(f, f^r) = 4$. Since, $f \in A_n$, using Proposition 5.2.4, we can consider that all the matching places of f have the value zero. Thus, there are two places in f^u which are not matching with two places in $(f^l)^r$. Also, $f \in A_n$ gives, $wt(f^u) = wt(f^l) = 1$. For $f \in A_n$, $wt(f) = 2$ iff f is a palindrome (i.e. $d(f, f^r) = 0$). Thus $d(f, f^r) \neq 4$. ∎

Proposition 5.4.2 *For $n \geq 3$ and $f \in A_n$, $d(f, f^r) \neq 2^n - 4$.*

Proof : Let, in contrary, $d(f, f^r) = 2^n - 4$. So, $M(f^u, (f^l)^r) = 2$. Since, $f \in A_n$, using Proposition 5.2.4, we can consider that the 4 places where f, f^r match, have the value zero. This selects $0 \leq y \neq z \leq 2^{n-1} - 1$, such that $f^u[y] = f^u[z] = 0$.

Let us consider the truth table of f. Except the leftmost input column, each of the input columns are of the form ll, where l is of length 2^{n-1}. (The leftmost input column in the truth table is 2^{n-1} 0's followed by 2^{n-1} 1's.) Note that, there exists at least one such l, so that $l[y] \neq l[z]$. In particular, we can choose $l[y] = 0, l[z] = 1$. (The proof using $l[y] = 1, l[z] = 0$ is similar.) Another important point is $l = l^{rc}$.

First we show $d(f^u, l) = 2^{n-2}$. Since, f is correlation immune, $wd(f, ll) = 0$. Now, $wd(f, ll) = wd(f^u f^l, ll) = wd(f^u, l) + wd(f^l, l) = wd(f^u, l) + wd((f^l)^{rc}, l^{rc})$. We have, $f^u[i] = (f^l)^{rc}[i]$, for $0 \leq i \leq 2^{n-1} - 1$, $i \neq y, i \neq z$, and $f^u[i] = (f^l)^r[i]$, for $i = y, z$. Also, $f^u[y] = l[y]$ *and* $f^u[z] \neq l[z]$. Now, $(f^l)^r[y] = f^u[y]$ and $l^r[y] \neq l[y]$ and $(f^l)^r[z] = f^u[z]$ and $l^r[z] \neq l[z]$. Thus, out of these 4 places (2 each for f^u, f^l) f and ll match at 2 places and do not match at 2 places. Thus, Walsh distance contribution from these four places is 0. The Walsh distance contribution from other $2^n - 4$ positions will also be 0. For $0 \leq i \leq 2^{n-1} - 1$, $i \neq y, i \neq z$ $f^u[i] = (f^l)^{rc}[i]$, and also we have, $l[i] = l^{rc}[i]$. Thus, either both the pair $f^u[i], l[i]$ and $(f^l)^{rc}[i], l^{rc}[i]$ match or both the pair do not match. Hence the Walsh distance contribution is same from both the top half and bottom half and each should be individually equal to zero. So we get, for $0 \leq i \leq 2^{n-1} - 1$, $i \neq y, i \neq z$, $\#(f^u[i] = l[i]) = \#(f^u[i] \neq l[i])$, which gives, $\#(f^u[i] \neq l[i]) = 2^{n-2} - 1$. Moreover, $f^u[y] = l[y]$ *and* $f^u[z] \neq l[z]$. Thus, $d(f^u, l) = 2^{n-2}$. Also, since f is correlation immune, $wt(f^u) = wt(f^l) = \frac{2^n - 4}{4} = 2^{n-2} - 1$.

Let, $a_0 = \#(f^u = 0, l = 0)$, $a_1 = \#(f^u = 0, l = 1)$, $a_2 = \#(f^u = 1, l = 0)$, $a_3 = \#(f^u = 1, l = 1)$. Since, $d(f^u, l) = 2^{n-2}$, $a_1 + a_2 = 2^{n-1}$. Also, $a_0 + a_1 = \#(f^u = 0) = 2^{n-1} - wt(f^u) =$

$2^{n-2}+1$. Similarly, $a_0 + a_2 = \#(l = 0) = 2^{n-1}$. Hence, $a_1 - a_2 = 1$. Now, $a_1 + a_2 = 2^{n-1}$ and $a_1 - a_2 = 1$ can't have any integer solution. Hence $d(f, f^r) \neq 2^n - 4$. ■

Let $DIST_n = \{d(f, f^r) \mid f \in A_n\}$. Then we get the following result.

Proposition 5.4.3 $\{y_1 + y_2 \mid y_1, y_2 \in DIST_n\} \subseteq DIST_{n+1}$.

Proof : Consider $f, g \in A_n$. Then the $n+1$ variable function $F = f^u g^u g^l f^l \in A_{n+1}$. Also, $d(F, F^r) = d(f, f^r) + d(g, g^r)$. Thus the proof. ■

We have checked by running computer program that $DIST_4 = \{0, 8, 16\}$. Thus from Proposition 5.4.3, $\{0, 8, 16, 24, 32\} \subseteq DIST_5$. However, there may very well be other values for $d(f, f^r)$ for $f \in A_5$. Note that from Proposition 5.4.1 and Proposition 5.4.2, it is clear that both 4 and 28 do not belong to $DIST_5$. Also from Proposition 5.2.2, we know that $d(f, f^r)$ is congruent to 0 mod 4. Thus the other two values we need to check for membership in $DIST_5$ are 12 and 20. By running computer program we have found functions $f \in A_5$ such that $d(f, f^r) = 12, 20$. That is, $DIST_5 = \{0, 8, 12, 16, 20, 24, 32\}$. Hence we get the following result which exactly characterizes what values can $d(f, f^r)$ take when f is correlation immune. The proof follows from Proposition 5.4.1, Proposition 5.4.2 and Proposition 5.4.3.

Theorem 5.4.1 *For* $n \geq 3$, $DIST_n = \{i \mid 0 \leq i \leq 2^n, i \equiv 0 \bmod 4, i \neq 4, 2^n - 4\}$.

Proof : For $n \leq 5$ the proof holds from the above discussion. Next we prove it by induction for $n > 5$. Let us consider it is true for all $n \leq m$. Thus, $DIST_m = \{i \mid 0 \leq i \leq 2^m, i \equiv 0 \bmod 4, i \neq 4, 2^m - 4\}$. Also, we have $\{y_1 + y_2 \mid y_1, y_2 \in DIST_m\} \subseteq DIST_{m+1}$. Thus, $\{i \mid 0 \leq i \leq 2^{m+1}, i \equiv 0 \bmod 4, i \neq 4, 2^{m+1} - 4\} \subseteq DIST_{m+1}$. From Proposition 5.4.1, Proposition 5.4.2 we have $4 \notin DIST_{m+1}$ and $2^{m+1} - 4 \notin DIST_{m+1}$. Hence, $\{i \mid 0 \leq i \leq 2^{m+1}, i \equiv 0 \bmod 4, i \neq 4, 2^{m+1} - 4\} = DIST_{m+1}$. ■

Also we get the following formula on number of correlation immune Boolean functions.

Theorem 5.4.2 $\mid A_n \mid = \sum_{i=0}^{2^{n-2}} 2^{2i} C_{n,4i}(2^{n-1} - 2i)$.

Proof : First we find out $f \in A_n$ with minimum weight such that $M(f, f^r) = 4i$. Using Proposition 5.2.4 we can consider that all the matching places have value 0. Now $d(f, f^r) = 2^n - 4i$. So, weight $wt(f) = \frac{2^n - 4i}{2} = 2^{n-1} - 2i$. Thus, $C_{n,4i}(2^{n-1} - 2i)$ is the number of functions where all the matching places are 0. Thus, for $f \in CIW_{n,4i}(2^{n-1} - 2i)$, there are $2i$ places in f^u, where we can choose any pattern and use same values in the palindromic positions in f^l. Thus, corresponding to each $f \in CIW_{n,4i}(2^{n-1} - 2i)$, we can construct 2^{2i} distinct functions. It is also clear that all such $2^{2i} C_{n,4i}(2^{n-1} - 2i)$ functions are distinct.

Thus, $\mid A_n \mid = \sum_{i=0}^{2^{n-2}} 2^{2i} C_{n,4i}(2^{n-1} - 2i)$. To be more specific, from Proposition 5.4.1 and Proposition 5.4.2 we have $C_{n,4i}(2^{n-1} - 2i) = 0$ for $4i = 2^n - 4, 4$. Hence, we get, $\mid A_n \mid = \sum_{i=0, i \neq 1, 2^{n-2}-1}^{2^{n-2}} 2^{2i} C_{n,4i}(2^{n-1} - 2i)$. ■

Though Theorem 5.4.2 provides a closed form expression, it seems time consuming to calculate that as $C_{n,4i}(2^{n-1} - 2i)$ need to be calculated for different i's. Similarly Theorem 5.3.1 and Theorem 5.3.2 provide formulae for $C_n(2a)$ and $\mid A_n \mid$ respectively. To use them, one has to determine the $C_{n,x_j}(2^{n-1})$'s. These are open problems and can prove to be nontrivial tasks. Also Theorem 5.3.2 shows that the enumeration problem of CI functions is related to the enumeration problem of balanced CI functions and the enumeration problem will be completely solved provided each partition of balanced CI functions can be calculated separately.

The correlation immune functions with same values of $M(f, f^r)$ form an equivalence class and in each equivalence class there is a hierarchy depending on $wt(f)$. Also, each set $CIW_n(2a)$ can be partitioned depending on $M(f, f^r)$. This gives a new direction to characterize and enumerate the correlation immune Boolean functions.

Chapter 6

Construction of Some Correlation Immune Functions

From the results of the previous chapter it is clear that though closed form expressions on the number of correlation immune functions can be achieved, getting exact enumeration by constructive methods is a nontrivial task. Here we concentrate on a different direction. We introduce new ideas to identify small but interesting subsets of correlation immune functions from construction point of view and find out the interrelationship among these subsets. We also estimate the cardinality of these subsets. The idea of these constructions will be used in the next chapter to find new classes of resilient Boolean functions with very good cryptographic properties.

Our contribution in this chapter is not in the direction of enumerative results. Currently best known enumerative result is provided in [28] which uses probabilistic techniques to give an asymptotic estimate. However, there are other enumeration results [80, 107, 118, 81] which uses necessary and sufficient conditions in construction of correlation immune functions. From enumeration point of view these results (and also our results) are not as good as [28], though these results have interesting constructive consequences. We here sharpen the necessary and sufficient conditions provided in [80, 107, 118, 81] to identify some small subsets of correlation immune functions. These results provide interesting combinatorial insight in the study of correlation immune functions.

6.1 Introduction

Construction and enumeration problems for cryptographically significant Boolean functions including correlation immunity was considered initially in [80, 107] and later in [28, 118, 81].

Following Mitchell [80], we define several important cryptographic properties of Boolean functions. The definitions are for a scalar valued Boolean function, since in most cases (except balancedness) the problem for vector valued Boolean function can be trivially reduced to that of the scalar valued one. In this chapter we follow the same notation as in Chapter 3. We consider the properties (C1) Balancedness, (C2) Nonaffinity, (C3) Nondegeneracy and (C4) Correlation Immunity. Also we denote the set of n variable Boolean functions which have the properties $Ci_1, \ldots, Ci_t$ by $A_n(i_1, \ldots, i_t)$. *The set of all correlation immune (CI) Boolean functions of n variables is denoted by A_n, i.e., $A_n = A_n(4)$.* The problems tackled in the papers [80, 107, 118, 81] are that of first order correlation immune functions, which we also continue here. That is, by correlation immunity we here mean correlation immunity of order 1. We also denote by $B_n = \Omega_n - A_n$, the set of all n variable non correlation immune (NCI) functions. Note that Denisov [28] has considered enumeration of higher order correlation immune functions too.

Next we briefly review the works related to construction of some subsets of correlation immune Boolean functions. A subset of A_n with cardinality $2^{2^{n-1}}$ was presented by Mitchell in [80]. This is basically the set of palindromic correlation immune Boolean functions. The other known subsets of A_n have the following cardinalities.

1. A subset of cardinality $2^{2^{n-1}} + 2^n - 2n + 2^{2^{n-4}} - 2^{n-3}$ was identified by Yang et al, in [118].
2. Another subset of size $|A_{n-1}|^2$ was proposed by Park et al in [81]. This has been done by recursive construction.

The recent results on the supersets of A_n using necessary conditions are as follows.

1. Yang and Guo [118] provided a superset of size $\sum_{k=0}^{2^{n-1}} \sum_{r=0}^{k} \binom{2^{n-2}}{r}^2 \binom{2^{n-2}}{k-r}^2$.
2. A superset of size $\sum_{j=0}^{2^{n-2}} \binom{2^{n-2}}{j}^4$ was obtained by Park et al [81].

Related results are also presented in [102, 26].

It is important to mention the work of Denisov [28] at this point. In [28] the following asymptotic formula for $N(n,k)$, the number of n-variable, k-th order correlation immune function has been obtained.

$$N(n,k) \sim \frac{2^{2^n}}{2^k \exp\{\sum_{i=1}^{k} (\ln \sqrt{\frac{\pi}{2}} + (\frac{n}{2} - i) \ln 2) \binom{n}{i}\}}.$$

For first order correlation immune functions, i.e. for $k = 1$,

$$A_n \sim \frac{2^{2^n}}{2\exp\{(\ln\sqrt{\frac{\pi}{2}} + (\frac{n}{2} - 1)\ln 2)n\}}.$$

The result in Denisov's paper is a probabilistic estimate which provides an asymptotic formula of $N(n,k)$. This result, when considered as an upper bound, presents the best known result in terms of enumeration.

Throughout this chapter $\subseteq$ denotes subset and $\subset$ denotes proper subset. Given the truth table of a function f of n input variables, we denote the output column of f in the truth table as f itself, i.e. we also interpret f as a binary string. We write $f = f^u f^l$, where f^u (respectively f^l) is the upper half (respectively lower half) of f. The strings f^r and f^c are respectively the reverse and bitwise complement of f. Further, $CIW_n(a) = \{f \in A_n \mid wt(f) = a\}$ and $NCIW_n(a) = \{f \in B_n \mid wt(f) = a\}$. These are required to denote the functions of same weight. We write $C_n(a) = \mid CIW_n(a) \mid$ and $N_n(a) = \mid NCIW_n(a) \mid$. By $C_n^k(a)$ we mean $(C_n(a))^k$. Note that $CIW_n(2^{n-1}) = A_n(1,4)$. In the previous chapter, it has been shown that $C_n(2a) < C_n(2a+2)$ for $2a < 2^{n-1}$.

6.2 Preliminary Results

In this section we start with a few important technical results.

Lemma 6.2.1

1. *$Prob(f = X_i) = \frac{1}{2}$ iff $\#(f = 1 \mid X_i = 0) = \#(f = 1 \mid X_i = 1)$.*
2. *$f \in A_n$ iff $d(f, X_i) = 2^{n-1}$, $\forall i,\ 1 \leq i \leq n$.*
3. *$f \in A_n$ iff $wd(f, X_i) = 0$, $\forall i,\ 1 \leq i \leq n$.*
4. *If $f \in A_n$, then $wt(f)$ is even.*
5. *If $wt(f)$ is odd, then f is non correlation immune, i.e. $CIW_n(2a+1) = \emptyset$.*

Item 3 of Lemma 6.2.1 is a simpler version of the Walsh transform characterization of first order CI functions (see [42]).

Proposition 6.2.1 *Consider $h_1, h_2 \in \Omega_{n-1}$, with $wt(h_1) = wt(h_2)$ and $f \in \Omega_n$ with $f = h_1 h_2$.*

1. *If both* $h_1, h_2 \in A_{n-1}$ *then* $f \in A_n$.

2. *If* $h_1 \in A_{n-1}$ *and* $h_2 \in B_{n-1}$ *then* $f \in B_n$.

3. *If* $h_1 = h_2^r$ *then* $f \in A_n$.

4. *If* $h_1 \in B_{n-1}$ *and* $h_1 = h_2$ *then* $f \in B_n$.

5. *If both* $h_1, h_2 \in B_{n-1}$, *with* $h_1 \neq h_2$ *and* $h_1 \neq h_2^r$, *then* f *may or may not belong to* A_n *(see Remark 6.2.1).*

6. *If* $f \in A_n$ *then either both* $h_1, h_2 \in A_{n-1}$ *or both* $h_1, h_2 \in B_{n-1}$.

Remark 6.2.1 *The main bottleneck in getting CI functions in constructive way is item 5 of Proposition 6.2.1, i.e., it is possible to concatenate two NCI functions of same weight and obtain both CI and NCI functions. We provide two such examples.*
Let $h_1 = 1000$ *and* $h_2 = 0100$, *where,* $h_1, h_2 \in B_2$ *and* $h_1 \neq h_2$, $h_1 \neq h_2^r$. *Let* $f \in \Omega_3$, *where,* $f = h_1 h_2 = 1000\ 0100$. *Then* $f \in B_3$, *i.e.,* $f \notin A_3$.
Let $h_1 = 10000100$ *and* $h_2 = 00010010$, *where,* $h_1, h_2 \in B_3$ *and* $h_1 \neq h_2$, $h_1 \neq h_2^r$. *Let* $f \in \Omega_4$, *where,* $f = h_1 h_2 = 10000100\ 00010010$. *Then* $f \in A_4$.
For a complete construction of all CI functions, it is necessary to identify when concatenation of two NCI functions of the same weight gives rise to a CI function. This, in general, is difficult. Here we provide partial solution to the problem.

Lemma 6.2.2 *Let* $f(X_n, \ldots, X_1)$ *be a Boolean function of* n *variables. Then* f *is CI iff for any* X_i, $1 \leq i \leq n$, $wt(f \& X_i) = wt(f \& X_i^c)$, *where* $S_1 \& S_2$ *is the bitwise AND of* S_1 *and* S_2.

Lemma 6.2.2 is another characterization of CI functions (see also Lemma 5 of [118]). Based on Lemma 6.2.2 we can use the principle of inclusion and exclusion to obtain an expression for $\mid A_n \mid$. Let $a_i = \{f \in \Omega_n \mid wt(f \& X_i) \neq wt(f \& X_i^c)\}$. and $\bar{a_i} = \Omega_n - a_i$. Let $N(a_i) = \mid a_i \mid$ and denote $a_i \cap a_j$ by $a_i a_j$. Then we get the following.

Theorem 6.2.1 $A_n = \bar{a_1} \cap \ldots \cap \bar{a_n}$ *and hence* $\mid A_n \mid = N(\bar{a_1} \ldots \bar{a_n}) = 2^{2^n} - \binom{n}{1} N(a_1) + \binom{n}{2} N(a_1 a_2) - \ldots + (-1)^n \binom{n}{n} N(a_1 \ldots a_n)$.

Proof : The first statement follows from Lemma 6.2.2. The second statement follows from the principle of inclusion and exclusion and noting that for any choice of $i_1, \ldots, i_r$ from $\{1, \ldots, n\}$, $N(a_{i_1} \ldots a_{i_r}) = N(a_1 \ldots a_r)$. ■

This expression seems difficult to handle, since it is complicated to evaluate $N(a_1 \ldots a_r)$ for arbitrary r. However, it can be checked that $N(a_1) = 2^{2^n} - \binom{2^n}{2^{n-1}}$.

We also explore use of generating functions to provide bounds for $C_k(a)$. Let, $g_k(x) = \sum_{a=0}^{2^k} C_k(a)x^a$ and $C(x) = \sum_{k\geq 0} g_k(x)y^k = \sum_{k\geq 0}\sum_{a=0}^{2^k} C_k(a)x^a y^k$.

Also, $h_k(x) = \sum_{a=0}^{2^k} N_k(a)x^a$ and $N(x) = \sum_{k>0} h_k(x)y^k = \sum_{k>0}\sum_{a=0}^{2^k} N_k(a)x^a y^k$ for NCI functions.

Proposition 6.2.2 *(1)* $C_k(a)+N_k(a) = \binom{2^k}{a}$, *(2)* $g_k(x)+h_k(x) = (1+x)^{2^k}$ *and (3)* $C(x)+N(x) = \sum_{k\geq 0}(1+x)^{2^k}y^k$.

Let $p(x) = \sum_{i\geq 0} p_i x^i$ and $q(x) = \sum_{i\geq 0} q_i x^i$. Define $p(x) \leq q(x)$ if $p_i \leq q_i$ for all $i \geq 0$. Also define $p^{[2]}(x) = \sum_{i\geq 0}(p_i x^i)^2$.

Lemma 6.2.3 $h_{n-1}(x)g_{n-1}(x) \leq h_n(x) \leq (1+x)^{2^n} - g_{n-1}^{[2]}(x)$.

Proof : From the Proposition 6.2.1 we get, $\sum_{r=0}^{a} N_{n-1}(r)C_{n-1}(a-r) \leq N_n(a) \leq \binom{2^n}{a} - C_{n-1}^2(\frac{a}{2})$. Note that, if a is odd or $\equiv 2 \bmod 4$ then $C_{n-1}(\frac{a}{2})$ is zero.

The result then follows from the fact that generating function for the convolution of two sequences is the product of the generating functions of the individual sequence. ■

In [118], it was commented that $\mid A_n \mid$ can be represented as $(2^{2^{n-1}})^{c_n}$, with c_n between 1 and 2. Since, $\mid A_n \mid > \mid A_{n-1} \mid^2$, c_n is strictly increasing. It is then interesting to find out whether this limit is strictly less than 2. Let $(2^{2^{n-1}})^{\sigma} = \sum_{j=0}^{2^{n-2}} \binom{2^{n-2}}{j}^4$, the upper bound on $\mid A_n \mid$ provided in [81]. Then $\lim_{n\to\infty}\sigma = 2$. However, the expression of the form $(2^{2^{n-1}})^{c_n}$ is not very useful in providing a clear estimation of the proportion of $\mid A_n \mid$ in $\mid \Omega_n \mid$. Let $(2^{2^{n-1}})^{\sigma} = \frac{2^{2^n}}{\tau 2^{f(n)}}$, τ constant, and $f(n)$ a polynomial of n. Then even if $\lim_{n\to\infty}\sigma = 2$, we get, $\lim_{n\to\infty}\left|\frac{A_n}{\Omega_n}\right| = 0$, giving the indication that the number of correlation immune Boolean functions is very few compared to the set of all Boolean functions. The result of [28] also gives the same indication. Using the result of [28] we get,

$$\lim_{n\to\infty}\frac{\mid A_n \mid}{\mid \Omega_n \mid} \sim \lim_{n\to\infty}\frac{1}{2\exp\{(\ln\sqrt{\frac{\pi}{2}} + (\frac{n}{2}-1)\ln 2)n\}} = 0.$$

6.3 Some Constructions

In this section we discuss different techniques to construct functions in A_n.

6.3.1 Basic Construction

Mitchell [80] has provided a subset of A_n with cardinality $2^{2^{n-1}}$ by showing that the set of Boolean functions, with the property that inverting the input leaves the output unchanged, is CI. We restate the same as follows. Both our interpretation and proof are simpler than that of [80].

Lemma 6.3.1 $D_n^1 = \{F \in \Omega_n \mid F \text{ palindrome}\} \subset A_n$. *Also,* $\mid D_n^1 \mid = 2^{2^{n-1}}$.

Proof : Let $f \in D_n^1$. For any column in the truth table corresponding to an input variable X_i and any $0 \leq \tau \leq 2^{n-1}-1$, $X_i[\tau] \neq X_i[2^n-1-\tau]$. This is due to the fact that if we look at any column, the locations at equal distance from top and bottom are always unequal. Let $f[\tau] = X_i[\tau]$. Then $f[\tau] \neq X_i[2^n-1-\tau]$. Let there be exactly d choices of τ in the first half of the X_i column where $f[\tau] = X_i[\tau]$. Hence in the second half there are exactly d choices of τ where $f[2^n-1-\tau] \neq X_i[2^n-1-\tau]$. Hence in the second half there are exactly $2^{n-1}-d$ choices of τ where $f[2^n-1-\tau] = X_i[2^n-1-\tau]$. Thus, if we consider the complete column X_i then there are $d + (2^{n-1}-d) = 2^{n-1}$ places where the column X_i and f matches. So $Prob(f = X_i) = \frac{1}{2}$. Thus, $f \in A_n$ proving $D_n^1 \subset A_n$. The proof that D_n^1 is a proper subset of A_n is clear from Lemma 6.3.2.

There are exactly $2^{2^{n-1}}$ palindromes of length 2^n. Hence, $\mid D_n^1 \mid = 2^{2^{n-1}}$. ∎

Next we show f may be CI even if f is not a palindrome. We define two sets in this direction. $D_n^2 = \{S_1 S_1^r S_2^r S_2 \mid wt(S_1) = wt(S_2),$ and $S_1 \neq S_2^r, S_1, S_2 \in \Omega_{n-2}\}$, $D_n^3 = \{SS^cS^cS \mid wt(S) \neq 2^{n-3}, S \in \Omega_{n-2}, S \neq S^r\}$.

Lemma 6.3.2 *(1)* D_n^1, D_n^2, D_n^3 *are disjoint proper subsets of* A_n, *(2)* $\mid D_n^2 \mid = \binom{2^{n-1}}{2^{n-2}} - 2^{2^{n-2}}$ *and (3)* $\mid D_n^3 \mid = 2^{2^{n-2}} - \binom{2^{n-2}}{2^{n-3}} - (2^{2^{n-3}} - \binom{2^{n-3}}{2^{n-4}})$.

Proof : First we prove that (1) holds. Since the elements of D_n^2 and D_n^3 are nonpalindromic, $D_n^1 \cap D_n^2 = \emptyset$ and $D_n^1 \cap D_n^3 = \emptyset$. Again in D_n^3, $wt(S) \neq wt(S^c)$ as $wt(S) \neq 2^{n-3}$, and hence $S \neq (S^c)^r$. Thus, $D_n^2 \cap D_n^3 = \emptyset$.

Now we prove (2). We choose an i such that $0 \leq i \leq 2^{n-2}$. So we can choose S_1 in $\binom{2^{n-2}}{i}$ ways. Depending on the choice of S_1, we can choose S_2 in $(\binom{2^{n-2}}{i} - 1)$ ways leaving

out S_1^r, since S_1 and S_2^r need to be distinct. Hence, $\mid D_n^2 \mid = \sum_{i=0}^{2^{n-2}} \binom{2^{n-2}}{i}\left(\binom{2^{n-2}}{i} - 1\right) = \sum_{i=0}^{2^{n-2}} \binom{2^{n-2}}{i}^2 - \sum_{i=0}^{2^{n-2}} \binom{2^{n-2}}{i} = \binom{2^{n-1}}{2^{n-2}} - 2^{2^{n-2}}$.

The proof of (3) is found by discarding the balanced functions and palindromic functions from the set of Ω_{n-2} for the choice of S. ■

Let $D_n^t = D_n^1 \cup D_n^2 \cup D_n^3$. Then the following theorem is immediate from Lemma 6.3.1 and Lemma 6.3.2.

Theorem 6.3.1 $\mid D_n^t \mid = 2^{2^{n-1}} + \binom{2^{n-1}}{2^{n-2}} - \binom{2^{n-2}}{2^{n-3}} - 2^{2^{n-3}} + \binom{2^{n-3}}{2^{n-4}}$.

6.3.2 Recursive Construction

In this subsection we provide construction methods for correlation immune functions which improves the result of [81]. Let us consider the following construction where $F \in \Omega_n$ and $f, g \in \Omega_{n-1}$.

$F(X_1, X_2, \ldots, X_n) = f(X_1, X_2, \ldots, X_{n-1})(1 \oplus X_{n-1})(1 \oplus X_n)$
$\oplus\, g(X_1, X_2, \ldots, X_{n-1})X_{n-1}(1 \oplus X_n) \oplus g(X_1, X_2, \ldots, X_{n-1})(1 \oplus X_{n-1})X_n$
$\oplus\, f(X_1, X_2, \ldots, X_{n-1})X_{n-1}X_n.$ (1)

Park et al [81] had shown that if f and g are CI ($f, g \in A_{n-1}$), then F given by (1) above is also CI. From this they obtained the inequality $\mid A_n \mid \geq \mid A_{n-1} \mid^2, n \geq 3$. We interpret the function F given in (1) as follows.

Proposition 6.3.1 *Let $f, g \in \Omega_{n-1}$ and let $F \in \Omega_n$ be a function given by $F = f^u g^l g^u f^l$. Then F is given by (1).*

This interpretation is more intuitive and allows us to generalize the construction procedure. The inequality $\mid A_n \mid \geq \mid A_{n-1} \mid^2$ depends on generation of $F \in A_n$ from $f, g \in A_{n-1}$. Take any 2 functions f and g (not necessarily distinct) from A_{n-1}, and form a function F as given in Proposition 6.3.1. Then $F \in A_n$ and the construction process is a bijection, so there are at least $\mid A_{n-1} \mid \times \mid A_{n-1} \mid$ correlation immune functions in A_n. Here we consider generalizations of the construction procedure given in Proposition 6.3.1. It has been proved in [81] that the construction in Proposition 6.3.1 yields correlation immune functions if both f and g are correlation immune. However, there are other possible ways of constructing $F \in A_n$ from correlation immune functions $f, g \in A_{n-1}$. The following two propositions provide constructions which are similar to Proposition 6.3.1.

Proposition 6.3.2 *Let $f, g \in A_{n-1}$ and $F = f^u g^u g^l f^l$. Then $F \in A_n$.*

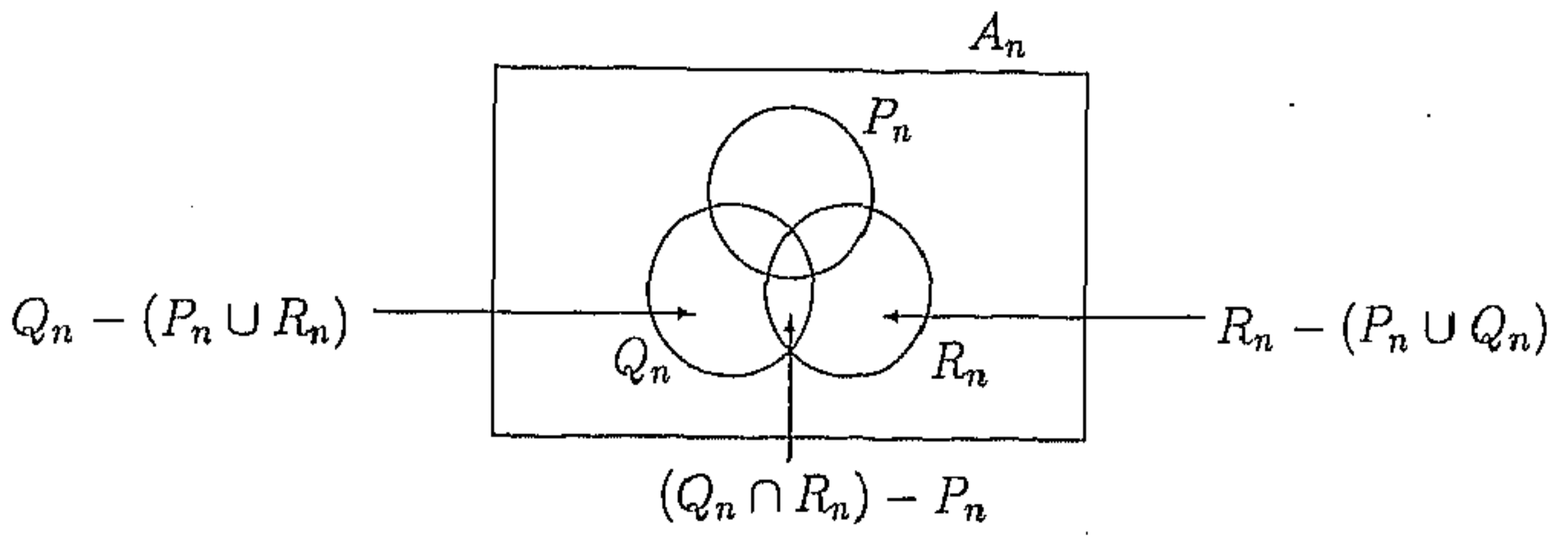


Figure 6.1: Venn Diagram

Proof : We show that $wd(F, X_i) = 0$ for all i. For $i \leq n-2$, $wd(F, X_i) = 0$ since f, g are CI. Also, $wd(F, X_j) = 0$, for $j = n-1, n$, holds since $wt(f^u) = wt(f^l)$ and $wt(g^u) = wt(g^l)$. Thus the result. ■

Proposition 6.3.3 *Let $f, g \in A_{n-1}$ and $wt(f) = wt(g)$. If $F = fg = f^u f^l g^u g^l$, then $F \in A_n$.*

Different possibilities similar to Propositions 6.3.1, 6.3.2, 6.3.3 are given in List 1.

List 1 : If $f, g \in A_{n-1}$, then $F \in A_n$ subject to the condition $wt(f) = wt(g)$, except for items 4,6,10 and 12, where $F \in A_n$ without the weight condition. Thus we can choose F from any of the following 12 constructions. However, Proposition 6.3.4 shows that all constructions of List 1 do not provide distinct sets.

1) $f^u f^l g^u g^l$ 2) $f^u f^l g^l g^u$ 3) $f^u g^u f^l g^l$ 4) $f^u g^u g^l f^l$ 5) $f^u g^l f^l g^u$ 6) $f^u g^l g^u f^l$
7) $f^l f^u g^u g^l$ 8) $f^l f^u g^l g^u$ 9) $f^l g^u f^u g^l$ 10) $f^l g^u g^l f^u$ 11) $f^l g^l f^u g^u$ 12) $f^l g^l g^u f^u$

Proposition 6.3.4 *Let f be a correlation immune function and $f = f^u f^l$. Let g be such that $g = g^u g^l = f^l f^u$, i.e. the top and bottom halves of the output string are interchanged. Then g is also correlation immune.*

Definition 6.3.1 *(1)* $P_n = \{f^u g^u g^l f^l \mid f, g \in A_{n-1}\}$. *(2)* $Q_n = \{f^u f^l g^u g^l \mid f, g \in A_{n-1},\ wt(f) = wt(g)\}$. *(3)* $R_n = \{f^u g^u f^l g^l \mid f, g \in A_{n-1},\ wt(f) = wt(g)\}$.

According to Proposition 6.3.4, in the List 1, items 4, 6, 10, 12 represent the same set P_n, items 1, 2, 7, 8 represent the same set Q_n, and items 3, 5, 9, 11 represent the same set R_n.

Consider $F \in Q_n \cup R_n$. Note that $wt(F) \equiv 0 \bmod 4$. Now, if $F \in P_n$, then $wt(F)$ is either '0 mod 4' or '2 mod 4'. For $F \in P_n$, $wt(F) \equiv 2 \bmod 4$ when exactly one of f, g is of weight '2 mod 4' and another is of weight '0 mod 4'.

Next we present results towards the enumeration of the sets P_n, Q_n, R_n. Let $A_n^t = P_n \cup Q_n \cup R_n$. We also show that A_n^t is a proper subset of A_n.

Proposition 6.3.5 *$P_n \subset A_n$, $Q_n \subset A_n, R_n \subset A_n$ and $(P_n \cup Q_n \cup R_n) \subset A_n$, for $n \geq 4$.*

Proof : That P_n, Q_n, R_n are proper subsets of A_n will be clear from the remaining part of the proof. Now we show that there are functions in A_n which are not in $A_n^t = P_n \cup Q_n \cup R_n$. Let $S \in B_{n-2}$, and S is not balanced (such functions exist for $n \geq 2$). Since, $SS, SS^c \in B_{n-1}$, $SS^cS^cS \notin (P_n \cup Q_n \cup R_n)$. However, SS^cS^cS is a function of the form $X_n \oplus g$, where $g \in \Omega_{n-1}$ and g is balanced. Thus, by Siegenthaler's construction [109, Section VI], $SS^cS^cS \in A_n$. Note that there are at least $2^{2^{n-2}} - \binom{2^{n-2}}{2^{n-3}} - \mid A_{n-2} \mid$ choices of unbalanced NCI functions S. ∎

In the following we obtain a lower bound on $\mid A_n^t \mid = \mid P_n \cup Q_n \cup R_n \mid$. We can describe $P_n \cup Q_n \cup R_n$ as disjoint union of four sets (see Figure 6.1).

$$P_n \cup Q_n \cup R_n = P_n \cup ((Q_n \cap R_n) - P_n) \cup (Q_n - (P_n \cup R_n)) \cup (R_n - (P_n \cup Q_n)).$$

For $n \geq 4$, $\mid A_n^t \mid = \mid P_n \mid + \mid (Q_n \cap R_n) - P_n \mid + \mid Q_n - (P_n \cup R_n) \mid + \mid R_n - (P_n \cup Q_n) \mid$. Now, $\mid P_n \mid = \mid A_{n-1} \mid^2$ (see also [118]). We find functions in the other three sets which were not given in [118]. First we find functions which belong to $(Q_n \cap R_n) - P_n$. We define the following sets for this purpose.

Definition 6.3.2 *Let $wt(S_1) = wt(S_2) = wt(S_3) = wt(S_4)$.*

1. $U_n = \{S_1S_2S_3S_4 \mid S_i \in B_{n-2}, 1 \leq i \leq 4,$
   and $S_1S_2, S_3S_4, S_1S_3, S_2S_4 \in A_{n-1}$, *and* $S_1S_4, S_2S_3 \in B_{n-1}\}$.

2. $V_n = \{SS^rS^rS \mid S \in B_{n-2}\}$.

Lemma 6.3.3 $U_n = (Q_n \cap R_n) - P_n$.

Proof : Let $F \in U_n$. Then from Definition 6.3.2, $F \in Q_n$, $F \in R_n$, $F \notin P_n$. Thus $U_n \subseteq (Q_n \cap R_n) - P_n$. On the other hand, let $F \in (Q_n \cap R_n) - P_n$. Now F can be written as $S_1S_2S_3S_4$, where $S_1, S_2, S_3, S_4 \in \Omega_{n-2}$. Since $S_1S_2 \in A_{n-1}$, either both $S_1, S_2 \in A_{n-2}$ or both $S_1, S_2 \in B_{n-2}$ (using Proposition 6.2.1). Similarly $S_1S_3 \in A_{n-1}$ forces either both $S_1, S_3 \in A_{n-2}$ or both $S_1, S_3 \in B_{n-2}$. Since, $S_2S_3 \in B_{n-1}$, both S_2 and S_3 can't be in A_{n-2}. So, both of them must be in B_{n-2} and hence S_1 is also in B_{n-2}. Similarly it can be shown that S_4 too belongs to B_{n-2}. Thus $F \in U_n$, which implies $(Q_n \cap R_n) - P_n \subseteq U_n$. ∎

Lemma 6.3.4 *$V_n \subseteq U_n$. Also, $| V_n | = | B_{n-2} | = 2^{2^{n-2}} - | A_{n-2} |$.*

Proof : By Lemma 6.3.1, if $f = SS^r$, $g = S^rS$ where $S \in B_{n-2}$, then $f, g \in A_{n-1}$. Thus $V_n \subseteq Q_n$ and also $V_n \subseteq R_n$.

Let $F \in V_n$ and if possible $F \in P_n$. As $F \in P_n$, F is of the form $f^u g^u g^l f^l$ where $f, g \in A_{n-1}$. However, $F \in V_n$, so F is of the form $F = SS^rS^rS$, where $S \in B_{n-2}$. Thus, $f = f^u f^l = SS$. Then by Proposition 6.2.1(item 4), $f \notin A_{n-1}$, which is a contradiction. Thus, we get $V_n \cap P_n = \emptyset$. and hence, $V_n \subseteq (Q_n \cap R_n) - P_n = U_n$. To get $| V_n |$ note that for S we can choose any function from B_{n-2}. ■

Next we clearly identify the relation between V_n and U_n.

Lemma 6.3.5 *$V_4 = U_4$ and $V_n \subset U_n$, $n \geq 5$.*

Proof : First we take $n = 4$. If we try to build $F \in U_4$, we have to start with $S_i \in B_2, 1 \leq i \leq 4$ of same weight. For S_i, S_j, $i \neq j$, we have, $S_iS_j \in B_3$ unless $S_i = S_j^r$. Hence $S_1S_2S_3S_4$ must be of the form SS^rS^rS. So, $V_4 = U_4$.

To show that $V_n \subset U_n$, $n \geq 5$, we will construct functions in $V_n - U_n$. Consider the case for $n = 5$. Let $S_1 = 10000100$ and $S_2 = 00010010$. Note that, both $S_1, S_2 \in B_3$. So, by item (4) of Proposition 6.2.1, $S_1S_1, S_2S_2 \in B_4$. However, $S_1S_2 \in A_4$. So, $S_1S_2S_2S_1 \in U_5$. Also, $S_1 \neq S_2^r$. Thus, $S_1S_2S_2S_1 \notin V_5$. Hence, $V_5 \subset U_5$.

For $n > 5$, say $n = 5+k$, $k > 0$, take $h_1 = S_1S_1 \ldots S_1$, (2^k times) and $h_2 = S_2S_2 \ldots S_2$, (2^k times). Thus, we have, $h_1, h_2 \in B_{3+k}$, $h_1 \neq h_2^r$ and $h_1h_2 \in A_{4+k}$ and also $h_1h_1, h_2h_2 \in B_{4+k}$. Thus, $h_1h_2h_2h_1 \notin V_n$. However, $h_1h_2h_2h_1 \in U_n$. This completes the proof. ■

The above lemma provides that $U_n - V_n \neq \emptyset$ for $n \geq 5$. Next we define a few sets to find functions of Q_n which do not belong to $(P_n \cup R_n)$.

Definition 6.3.3 *Let $wt(S_1) = wt(S_2) = wt(S_3) = wt(S_4)$.*

1. $Q_n^x = \{S_1S_2S_3S_4 \mid S_1, S_2 \in B_{n-2}, S_1S_2 \in A_{n-1}, S_3, S_4 \in A_{n-2}\}$
   $\cup \{S_1S_2S_3S_4 \mid S_3, S_4 \in B_{n-2}, S_3S_4 \in A_{n-1}, S_1, S_2 \in A_{n-2}\}$

2. $Q_n^y = \{S_1S_2S_3S_4 \mid S_i \in B_{n-2}, 1 \leq i \leq 4,$ *and* $S_1S_2, S_3S_4 \in A_{n-1},$
   and $S_1S_3, S_1S_4, S_2S_3, S_2S_4 \in B_{n-1}\}$

3. $Q_n^0 = Q_n^x \cup Q_n^y$

4. $Q_n^1 = \{S_1S_1^rS_3S_4, S_3S_4S_1S_1^r \mid S_1 \in B_{n-2}, S_3, S_4 \in A_{n-2}\}$

Lemma 6.3.6 *(1)* $Q_n^0 = Q_n - (P_n \cup R_n)$, *(2)* $Q_4^1 = Q_4^x$, $Q_n^1 \subset Q_n^x$ *for* $n \geq 5$, *(3)* $| Q_n^1 | = 2 \sum_{r=1}^{2^{n-3}-1} C_{n-2}^2(2r) N_{n-2}(2r)$.

Proof : Statement (1) can be proved in the same way as Lemma 6.3.3 and (2) can be proved similar to Lemma 6.3.5. Next we prove (3). Let us consider the form $S_1 S_1^r S_3 S_4$. Since any correlation immune function is of even weight, we only consider the even weight functions of Ω_{n-2}. Also, there is no function in B_{n-2} of weight 0 or 2^{n-2}. Thus we only consider the functions of Ω_{n-2} of weight $2r$, where r varies from 1 to $2^{n-3}-1$. Now S_3 and S_4 can be any two correlation immune function, and so can be chosen in $C_{n-2}^2(2r)$ ways, whereas S_1 can be chosen in $N_{n-2}(2r)$ ways. Thus we get the choice of $\sum_{r=1}^{2^{n-3}-1} C_{n-2}^2(2r) N_{n-2}(2r)$ functions. Now, the functions of the form $S_1 S_1^r S_3 S_4$ and $S_3 S_4 S_1 S_1^r$ are distinct, since the first one starts with a function from B_{n-2} whereas the second one starts with a function from A_{n-2}. Thus, we get the cardinality of Q_n^1. ■

Lemma 6.3.7 $Q_n^y \neq \emptyset$, $n \geq 4$.

Proof : Consider $F = S_1 S_1^r S_2 S_2^r$, where $S_1, S_2 \in B_{n-2}$, $wt(S_1) = wt(S_2)$, and both $S_1 S_2, S_1 S_2^r \in B_{n-1}$. Thus, $F \notin (P_n \cup R_n)$. Also $F \notin Q_n^x$, since all $S_1, S_1^r, S_2, S_2^r \in B_{n-2}$. Now, we have to show the existence of such $S_1, S_2 \in B_{n-2}$. Consider, $wt(S_1) = wt(S_2) = 1$, S_1 is of the form $1000\ldots0$ and S_2 is of the form $0100\ldots0$. Let $S_1 S_2, S_1 S_2^r$ be functions of input variables $X_1, \ldots, X_{n-1}$. Then $\#(S_1 S_2 = 1 \mid X_{n-2} = 0) = 2$ and $\#(S_1 S_2 = 1 \mid X_{n-2} = 1) = 0$. Thus by Lemma 6.2.1, $Prob(S_1 S_2 = X_{n-2}) \neq \frac{1}{2}$. Also, $\#(S_1 S_2^r = 1 \mid X_1 = 0) = 2$ and $\#(S_1 S_2^r = 1 \mid X_1 = 1) = 0$. Thus by Lemma 6.2.1, $Prob(S_1 S_2 = X_1) \neq \frac{1}{2}$. Hence, both $S_1 S_2, S_1 S_2^r \in B_{n-1}$. ■

Enumeration of Q_4^y is easy, since if we take any two $S_1, S_2 \in B_2$ of same weight with $S_1 \neq S_2$ and $S_1 \neq S_2^r$, we get both $S_1 S_2, S_1 S_2^r \in B_3$. However, the characterization for $n \geq 5$ is complicated. Next we find functions of $R_n - (P_n \cup Q_n)$.

Definition 6.3.4 *Let* $wt(S_1) = wt(S_2) = wt(S_3) = wt(S_4)$.

1. $R_n^x = \{S_1 S_2 S_3 S_4 \mid S_1, S_3 \in B_{n-2}, S_1 S_3 \in A_{n-1}, S_2, S_4 \in A_{n-2}\}$
   $\cup \{S_1 S_2 S_3 S_4 \mid S_2, S_4 \in B_{n-2}, S_2 S_4 \in A_{n-1}, S_1, S_3 \in A_{n-2}\}$
2. $R_n^y = \{S_1 S_2 S_3 S_4 \mid S_i \in B_{n-2}, 1 \leq i \leq 4,$ *and* $S_1 S_3, S_2 S_4 \in A_{n-1},$
   and $S_1 S_2, S_3 S_4, S_2 S_3, S_1 S_4 \in B_{n-1}\}$
3. $R_n^0 = R_n^x \cup R_n^y$

4. $R_n^1 = \{S_1 S_3 S_1^r S_4, S_3 S_1 S_4 S_1^r \mid S_1 \in B_{n-2}, S_3, S_4 \in A_{n-2}\}$

Remark 6.3.1 *Lemma 6.3.6 and Lemma 6.3.7 proved for Q are also true for R.*

Thus, from Lemma 6.3.5, Lemma 6.3.6, Lemma 6.3.7 and Remark 6.3.1 we get the following theorem.

Theorem 6.3.2 *(1)* $P_n \cup Q_n \cup R_n = P_n \cup U_n \cup Q_n^0 \cup R_n^0$ *for* $n \geq 4$.
(2) $P_4 \cup Q_4 \cup R_4 = P_4 \cup V_4 \cup Q_4^1 \cup Q_4^y \cup R_4^1 \cup R_4^y$. *(3) For* $n \geq 5$,
$| A_n | > | P_n \cup Q_n \cup R_n | > | P_n | + | U_n | + | Q_n^1 | + | Q_n^y | + | R_n^1 | + | R_n^y |$.

Proof : It is easy to see that (1) and (3) holds. The proof of (2) holds since $Q_4^x = Q_4^1$ and $R_4^x = R_4^1$, similar to Lemma 6.3.5. ■

Remark 6.3.2 *The above theorem suggests that exact enumeration of $P_n \cup Q_n \cup R_n$ for $n > 4$, is difficult using this kind of technique. It is clear from Lemma 6.3.5, Lemma 6.3.6 and Lemma 6.3.7 that the difficulty arises due to the problem of two NCI functions giving rise to a CI function (see Remark 6.2.1).*

Theorem 6.3.3 $| A_n^t | > | A_{n-1} |^2 + 2^{2^{n-2}} - | A_{n-2} | + 4 \sum_{r=1}^{2^{n-3}-1} C_{n-2}^2(2r)\ N_{n-2}(2r)$, $n \geq 4$.

Proof : Using Lemma 6.3.6, Theorem 6.3.2 and Remark 6.3.1,
$| P_n \cup Q_n \cup R_n | > | A_{n-1} |^2 + 2^{2^{n-2}} - | A_{n-2} | + 4 \sum_{r=1}^{2^{n-3}-1} C_{n-2}^2(2r) N_{n-2}(2r)$. ■

Lemma 6.3.8 $\sum_{r=1}^{2^{n-3}-1} C_{n-2}^2(2r)\ N_{n-2}(2r) \geq \sum_{r=1}^{2^{n-3}-1} \binom{2^{n-3}}{r}^2 \left(\binom{2^{n-2}}{2r} - \binom{2^{n-3}}{r}\right)$, $n \geq 4$.

Proof : $C_{n-2}(2r) + N_{n-2}(2r) = \binom{2^{n-2}}{2r}$, a constant for fixed r, $1 \leq r \leq 2^{n-3} - 1$. Now, $C_{n-2}^2(2r)\ N_{n-2}(2r)$ is increasing in $0 \leq C_{n-2}(2r) \leq \frac{2}{3}\binom{2^{n-2}}{2r}$. Also, $\binom{2^{n-3}}{r} \leq C_{n-2}(2r) \leq \frac{2}{3}\binom{2^{n-2}}{2r}$. So, $C_{n-2}^2(2r)\ N_{n-2}(2r) \geq \binom{2^{n-3}}{r}^2 \left(\binom{2^{n-2}}{2r} - \binom{2^{n-3}}{r}\right)$. Thus,
$\sum_{r=1}^{2^{n-3}-1} C_{n-2}^2(2r)\ N_{n-2}(2r) \geq \sum_{r=1}^{2^{n-3}-1} \binom{2^{n-3}}{r}^2 \left(\binom{2^{n-2}}{2r} - \binom{2^{n-3}}{r}\right)$. ■

Corollary 6.3.1 $\sum_{r=1}^{2^{n-3}-1} C_{n-2}^2(2r)\ N_{n-2}(2r) > \frac{2^{2^{n-1}}}{2^{\frac{3n}{2}-3}}$.

Proof : To get the proof, it should be noted that $\binom{2^{n-2}}{2^{n-3}}\binom{2^{n-3}}{2^{n-4}}^2 > 8\,\frac{2^{2^{n-1}}}{2^{\frac{3n}{2}}}$, by using Stirling's approximation $k! = \sqrt{2\pi k}\,(\frac{k}{e})^k$. ■

The closed form lower bound in Lemma 6.3.8 and Corollary 6.3.1 (though a very conservative estimate) clearly indicates that though the proportion of correlation immune functions in the total set of Boolean functions is negligible, in absolute sense, it is possible to find out a lot of such functions using deterministic construction methods.

6.4 Some necessary Conditions

The currently best known upper bound is the asymptotic formula provided by Denisov [28] which is as follows. $A_n \sim \frac{2^{2^n}}{2\exp\{(\ln\sqrt{\frac{\pi}{2}} + (\frac{n}{2}-1)\ln 2)n\}}$. This is a probabilistic estimate. Also there are some deterministic estimates which we mention below [118, 81]. The result of [28] is much sharper than the results of [118, 81]. However, the results of [118, 81] are also of interest since they provide some necessary conditions on the correlation immune Boolean functions.

The upper bound on $\mid A_n \mid$ was provided in [118] and it was later improved to $\mid A_n \mid \leq \sum_{j=0}^{2^{n-2}} \binom{2^{n-2}}{j}^4$ in [81]. This was obtained by showing that

$$A_n \subseteq K_n = \bigcup_{\mid j \mid=0}^{2^{n-3}} \{f_1 g_2 g_1 f_2 \mid \text{ where } f = f_1 f_2,\ g = g_1 g_2, \text{ and } f \in Y_{2j} \text{ and } g \in Y_{-2j}\},$$

where,

$$Y_j = \{f \in \Omega_{n-1} \mid \#(f = X_{n-1}) = \#(f \neq X_{n-1}),\ \#(f = X_{n-2}) - \#(f \neq X_{n-2}) = 2j\},$$

$\mid j \mid \leq 2^{n-2}$.

The above condition is necessary for a function to be correlation immune. It has been shown in [81] that $F = f^u g^l g^u f^l \in \Omega_n$ is CI, iff $wd(f^u f^l, X_{n-1}) = 0, wd(g^u g^l, X_{n-1}) = 0$ and $wd(f^u f^l, X_i) = -wd(g^u g^l, X_i)$ for $1 \leq i \leq n-2$. However, the following characterization holds,

Theorem 6.4.1 *$F = f^u g^u g^l f^l \in \Omega_n$, is CI, iff $wd(f^u f^l, X_{n-1}) = 0, wd(g^u g^l, X_{n-1}) = 0$ and for $1 \leq i \leq n-2$, $wd(f^u f^l, X_i) = -wd(g^u g^l, X_i) \equiv 0 \bmod 4$.*

The equivalence to '0 mod 4' in the above theorem was proved for only $i = n-2$ in [81]. The upper bound in [81] considered only three leftmost variables X_n, X_{n-1}, X_{n-2} in the truth table. One can get better necessary conditions by including variables X_i for $i < n-2$. However, if we consider the leftmost four variables then the conditions become complicated. So here we take a different approach. We show that there are functions in K_n which are not correlation immune. First we require the following two results.

Proposition 6.4.1 *Let $F \in \Omega_n$ be of the form $F = F_0F_1F_2F_3F_4F_5F_6F_7$. If k $(0 \leq k \leq 7)$ of the F_i's are CI functions and the other $8-k$ F_i's are equal to a NCI function, then F is NCI.*

Proposition 6.4.2 *Let $f, g \in \Omega_{n-1}$, with $f = f_1f_2f_3f_4$, and $g = g_1g_2g_3g_4$, where $wt(f_1) = wt(g_2) = a_1$, $wt(f_2) = wt(g_1) = b_1$, $wt(f_3) = wt(g_4) = a_2$, $wt(f_4) = wt(g_3) = b_2$, and $a_1 + b_1 = a_2 + b_2$. Then $F = f_1f_2g_3g_4g_1g_2f_3f_4 \in K_n$.*

The above two Propositions together constitute a sufficient condition for a function f to belong to $K_n - A_n$. So the problem reduces to constructing functions satisfying Proposition 6.4.1 and Proposition 6.4.2.

Theorem 6.4.2 $| K_n - A_n | \geq \sum_{k=0}^{7} \binom{8}{k} \sum_{a=0}^{2^{n-3}} C_{n-3}^{k}(a) N_{n-3}(a)$.

Proof : We consider functions of Ω_{n-3} of same weight. The conditions of Proposition 6.4.1, 6.4.2 are satisfied if one chooses k functions from $CIW_{n-3}(a)$ and 1 function from $NCIW_{n-3}(a)$, where k is as in Proposition 6.4.1. ∎

Remark 6.4.1 *We choose only one function from $NCIW_{n-3}(a)$ and repeat it in $8-k$ places instead of using $8-k$ possibly different functions from $NCIW_{n-3}(a)$. This is because concatenation of two different NCI functions may generate a CI function (see also Remarks 6.2.1, 6.3.2).*

Next we go for a more detailed analysis of the necessary conditions. Depending on the value of k in Proposition 6.4.1 several cases arise.
Cases C_0 to C_3 (corresponding to the $k = 0$ to $k = 3$) : If $a_1 = a_2 = b_1 = b_2$, then $F \in K_n - A_n$.

For $k \geq 4$, the situation is more complicated. Let $l_1 = \{f_1, g_2\}$, $l_2 = \{f_2, g_1\}$, $l_3 = \{f_3, g_4\}$, $l_4 = \{f_4, g_3\}$. The k CI functions are to be chosen from the sets l_1, l_2, l_3, l_4. Suppose we choose k_i functions from l_i, with $k_1 + k_2 + k_3 + k_4 = k$. The next tables different possible values of the k_i's and a corresponding condition on a_1, a_2, b_1 and b_2. If f and g are chosen such that this condition is satisfied, then the corresponding F is in K_n and is not in A_n.

Case C_4 (for $k = 4$)

k_1	k_2	k_3	k_4	condition
0	0	2	2	$a_1 = b_1 = a,\ a_2 + b_2 = 2a$
0	1	1	2	$a_1 = b_1 = a_2 = b_2 = a$
0	1	2	1	$a_1 = b_1 = a_2 = b_2 = a$
0	2	0	2	$a_1 = a_2 = a,\ b_1 = b_2 = b$
0	2	1	1	$a_1 = b_1 = a_2 = b_2 = a$
0	2	2	0	$a_1 = b_2,\ b_1 = a_2$
1	0	1	2	$a_1 = b_1 = a_2 = b_2 = a$
1	0	2	1	$a_1 = b_1 = a_2 = b_2 = a$
1	1	0	2	$a_1 = b_1 = a_2 = b_2 = a$
1	1	1	1	$a_1 = b_1 = a_2 = b_2 = a$
1	1	2	0	$a_1 = b_1 = a_2 = b_2 = a$
1	2	0	1	$a_1 = b_1 = a_2 = b_2 = a$
1	2	1	0	$a_1 = b_1 = a_2 = b_2 = a$
2	0	0	2	$a_1 = b_2,\ b_1 = a_2$
2	0	1	1	$a_1 = b_1 = a_2 = b_2 = a$
2	0	2	0	$a_1 = a_2 = a,\ b_1 = b_2 = b$
2	1	0	1	$a_1 = b_1 = a_2 = b_2 = a$
2	1	1	0	$a_1 = b_1 = a_2 = b_2 = a$
2	2	0	0	$a_2 = b_2 = a,\ a_1 + b_1 = 2a$

Case C_5 (for $k = 5$)

k_1	k_2	k_3	k_4	condition
0	1	2	2	$a_1 = b_1 = a,\ a_2 + b_2 = 2a$
0	2	1	2	$a_1 = a_2 = a,\ b_1 = b_2 = b$
0	2	2	1	$a_1 = b_2,\ b_1 = a_2$
1	0	2	2	$a_1 = b_1 = a,\ a_2 + b_2 = 2a$
1	1	1	2	$a_1 = b_1 = a_2 = b_2 = a$
1	1	2	1	$a_1 = b_1 = a_2 = b_2 = a$
1	2	0	2	$a_2 = a_1 = a,\ b_1 = b_2 = b$
1	2	1	1	$a_1 = b_1 = a_2 = b_2 = a$
1	2	2	0	$a_1 = b_2,\ b_1 = a_2$
2	0	1	2	$a_1 = b_2,\ b_1 = a_2$
2	0	2	1	$b_1 = b_2 = b,\ a_1 = a_2 = a$
2	1	0	2	$a_1 = b_2,\ b_1 = a_2$
2	1	1	1	$a_1 = b_1 = a_2 = b_2 = a$
2	1	2	0	$b_1 = b_2 = b,\ a_1 = a_2 = a$
2	2	0	1	$a_2 = b_2 = a,\ a_1 + b_1 = 2a$
2	2	1	0	$a_2 = b_2 = a,\ a_1 + b_1 = 2a$

Case C_6 (for $k = 6$)

k_1	k_2	k_3	k_4	condition
0	2	2	2	$a_1 + b_1 = a_2 + b_2$
1	1	2	2	$a_1 = b_1 = a,\ a_2 + b_2 = 2a$
1	2	1	2	$a_1 = a_2,\ b_1 = b_2$
1	2	2	1	$a_1 = b_2,\ b_1 = a_2$
2	0	2	2	$a_1 + b_1 = a_2 + b_2$
2	1	1	2	$a_1 = b_2,\ b_1 = a_2$
2	1	2	1	$b_1 = b_2,\ a_1 = a_2$
2	2	0	2	$a_1 + b_1 = a_2 + b_2$
2	2	1	1	$a_2 = b_2 = a,\ a_1 + b_1 = 2a$
2	2	2	0	$a_1 + b_1 = a_2 + b_2$

Case C_7 (for $k = 7$)

k_1	k_2	k_3	k_4	condition
1	2	2	2	$a_1 + b_1 = a_2 + b_2$
2	1	2	2	$a_1 + b_1 = a_2 + b_2$
2	2	1	2	$a_1 + b_1 = a_2 + b_2$
2	2	2	1	$a_1 + b_1 = a_2 + b_2$

Let S_k be the number of functions satisfying condition C_k, for $0 \leq k \leq 7$. Also we use two notations for expressing the formulae in short. Let $c(a)$ be the number of CI functions of $(n-3)$ variables of weight a and $n(a)$ be the number of NCI functions of $(n-3)$ variables of weight a. Then

$S_0 = \sum_{a=0}^{2^{n-3}} n(a) = \mid B_{n-3} \mid = 2^{2^{n-3}} - \mid A_{n-3} \mid$.

$S_1 = \sum_{a=0}^{2^{n-3}} \binom{8}{1} c(a)n(a)$. $S_2 = \sum_{a=0}^{2^{n-3}} \binom{8}{2} c^2(a)n(a)$. $S_3 = \sum_{a=0}^{2^{n-3}} \binom{8}{3} c^3(a)n(a)$.

Next we discuss the expressions for S_4, S_5, S_6, S_7. We discuss the result for S_4 in detail, the others being similar.

- Note the case C_4 in the table, where we have 19 rows.
- Now there are 13 rows with the condition $a_1 = b_1 = a_2 = b_2 = a$.
  - In 12 of those cases, the values are permutations of $0, 1, 1, 2$. The values $1, 1$ for two different k_i's provide $\binom{2}{1} \times \binom{2}{1} = 4$ ways to choose two CI functions from two different l_i sets.
  - In the case with the values $1, 1, 1, 1$ there are 4 options.

  These contribute $56 \times \sum_{a=0}^{2^{n-3}} c^4(a)n(a)$.
- There are 4 cases where two values are equal such as $a_1 = a_2 = a, b_1 = b_2 = b$ and these contribute $4 \times \sum_{a_1,a_2=0}^{2^{n-3}} c^4(a_1)n(a_2)$.
- There are two cases similar to $a_1 = b_1 = a, a_2 + b_2 = 2a$ and these contribute

$$2 \times \sum_{\substack{a_2, b_2, a = 0, \\ a_2 + b_2 = 2a,}}^{2^{n-3}} c^2(a_2)c^2(b_2)n(a).$$

Thus we get $S_4 = 56 \times \sum_{a=0}^{2^{n-3}} c^4(a)n(a) + 4 \times \sum_{a_1,a_2=0}^{2^{n-3}} c^4(a_1)n(a_2) + 2 \times \sum_{\substack{a_2, b_2, a = 0, \\ a_2 + b_2 = 2a,}}^{2^{n-3}} c^2(a_2)c^2(b_2)n(a).$

Similarly,

$$S_5 = 8 \times \sum_{\substack{a, a_2, b_2 = 0, \\ a_2 + b_2 = 2a}}^{2^{n-3}} c^2(a_2)c^2(b_2)c(a)n(a) + 16 \times \sum_{a,b=0}^{2^{n-3}} c^4(a)n(b)c(b) + 32 \times \sum_{a=0}^{2^{n-3}} c^5(a)n(a).$$

$$S_6 = 4 \times \sum_{\substack{a_1, a_2, a_3, a_4 = 0, \\ a_1 + a_2 = a_3 + a_4}}^{2^{n-3}} c^2(a_1)c^2(a_2)c^2(a_3)n(a_4) + 8 \times \sum_{\substack{a, a_1, a_2 = 0, \\ a_1 + a_2 = 2a}}^{2^{n-3}} c^2(a_1)c^2(a_2)c^2(a)n(a)$$
$$+ 16 \times \sum_{a_1,a_2=0}^{2^{n-3}} c^4(a_1)c^2(a_2)n(a_2).$$

$$S_7 = 8 \times \sum_{\substack{a_1, a_2, a_3, a_4 = 0, \\ a_1 + a_2 = a_3 + a_4}}^{2^{n-3}} c^2(a_1)c^2(a_2)c^2(a_3)c^2(a_4)n(a_4).$$

So we get the following result. Note that the sums involved are complicated and it seems difficult to simplify them.

Theorem 6.4.3 $\mid K_n - A_n \mid \geq \sum_{i=0}^{7} S_i$.

6.5 More than One Conditions

Here we provide functions in $A_n(1,2,3,4)$. First we consider the set $A_n(1,4)$. Note that a large proportion of functions in $A_n(1,4)$ are in $A_n(1,2,3,4)$. Theorem 6.5.2 is important in this direction. The result shows (using Theorem 6.5.1, Theorem 6.5.3) the proportion of $\mid A_n(1,2,3,4) \mid$ in $\mid A_n(1,4) \mid$ is almost equal to 1 for large n.
Initially we consider the following sets similar to Subsection 6.3.1.
(1) $D_n^{1b} = \{F \in \Omega_n \mid wt(F) = 2^{n-1},\ F \text{ is a palindrome }\}$,
(2) $D_n^{2b} = \{S_1 S_1^r S_2^r S_2 \mid wt(S_1) = wt(S_2) = 2^{n-3}, \text{ and } S_1 \neq S_2^r,\ S_1, S_2 \in \Omega_{n-2}\}$,
and (3) $D_n^{3b} = \{SS^cS^cS \mid wt(S) \neq 2^{n-3}, S \in \Omega_{n-2}, S \neq S^r\}$.

Theorem 6.5.1 *$D_n^{1b}, D_n^{2b}, D_n^{3b}$ are all proper subsets of $A_n(1,4)$. Also,*
$\mid D_n^{1b} \cup D_n^{2b} \cup D_n^{3b} \mid = \binom{2^{n-1}}{2^{n-2}} + 2^{2^{n-2}} + \binom{2^{n-2}}{2^{n-3}}\left(\binom{2^{n-2}}{2^{n-3}} - 2\right) - 2^{2^{n-3}} + \binom{2^{n-3}}{2^{n-4}}$.

Proof : Similar to the proof of Lemma 6.3.2, it can be checked that $D_n^{1b}, D_n^{2b}, D_n^{3b}$ are mutually disjoint subsets of $CIW_n(2^{n-1})$. Now it can be checked that $\mid D_n^{1b} \mid = \binom{2^{n-1}}{2^{n-2}}$, $\mid D_n^{2b} \mid = \binom{2^{n-2}}{2^{n-3}}\left(\binom{2^{n-2}}{2^{n-3}} - 1\right)$, and $\mid D_n^{3b} \mid = 2^{2^{n-2}} - \binom{2^{n-2}}{2^{n-3}} - (2^{2^{n-3}} - \binom{2^{n-3}}{2^{n-4}})$. ∎

Theorem 6.5.2 $\mid A_n(1,2,3,4) \mid = \mid A_n(1,3,4) \mid - 2 \geq \mid A_n(1,4) \mid - n \mid A_{n-1}(1,4) \mid - 2$.

Proof : The equality comes from existence of only two affine functions in $A_n(1,3,4)$. The inequality comes from the fact that there are at most $n \mid A_{n-1}(1,4) \mid$ degenerate functions in $A_n(1,4)$. ∎

Similar to Theorem 6.5.2, one can show $\mid A_n(3,4) \mid \geq \mid A_n(4) \mid (1 - \frac{n}{2^{2^{n-2}}})$ and so the enumeration problem for $A_n(3,4)$ is really that of $A_n(4)$. It is also interesting to see that all the functions of $CIW_n(2a)$, where a is odd, are nondegenerate. Now we use the techniques of Subsection 6.3.2 to provide recursive construction procedures.

Let us denote $T_n^\alpha = \{f^u g^u g^l f^l \mid f \in CIW_{n-1}(2a), g \in CIW_{n-1}(2^{n-1} - 2a),\ 0 \leq a \leq 2^{n-2},\ a \neq 2^{n-3}\}$ and $P_n^\alpha = \{f^u g^u g^l f^l \mid f, g \in A_{n-1}(1,4)\}$.

Proposition 6.5.1 *(1)* T_n^α, P_n^α *are mutually disjoint subsets of* $A_n(1,4)$.
(2) $P_n^\alpha = \mid A_{n-1}(1,4) \mid^2$. *(3)* $\mid T_n^\alpha \mid = 2 \sum_{i=0}^{2^{n-3}-1} C_{n-1}^2(2i)$, *for* $n \geq 4$.

Proof : Using Proposition 6.3.2, $T_n^\alpha, P_n^\alpha \subseteq A_n(4)$. Also it can be checked that $T_n^\alpha, P_n^\alpha \subseteq A_n(1,4)$ and $T_n^\alpha \cap P_n^\alpha = \emptyset$.

The proof of (2) is clear from the definition of P_n^α. The proof of (3) is derived from $C_{n-1}(2i) = C_{n-1}(2^{n-1} - 2i)$, since $f \in A_n$ iff $f^c \in A_n$. ∎

Next we construct a few sets in the same way as in Subsection 6.3.2.
$V_n^\alpha = \{SS^rS^rS \mid S \in \Omega_{n-2} - A_{n-2}(1,4), wt(S) = 2^{n-3}\}$,
$Q_n^{1\alpha} = \{S_1S_1^rS_3S_4, S_3S_4S_1S_1^r \mid S_1 \in \Omega_{n-2} - A_{n-2}(1,4), S_3, S_4 \in A_{n-2}(1,4),$
and $wt(S_1) = wt(S_3) = wt(S_4)\}$ and $R_n^{1\alpha} = \{S_1S_3S_1^rS_4, S_3S_1S_4S_1^r$
$\mid S_1 \in \Omega_{n-2} - A_{n-2}(1,4), S_3, S_4 \in A_{n-2}(1,4),$ and $wt(S_1) = wt(S_3) = wt(S_4)\}$.

Proposition 6.5.2 *(1)* $T_n^\alpha, P_n^\alpha, V_n^\alpha, Q_n^{1\alpha}, R_n^{1\alpha}$ *are mutually disjoint subsets of* $A_n(1,4)$,
(2) $\mid V_n^\alpha \mid = \binom{2^{n-2}}{2^{n-3}} - \mid A_{n-2}(1,4) \mid$,
(3) $\mid Q_n^{1\alpha} \mid = \mid R_n^{1\alpha} \mid = 2 \mid A_{n-2}(1,4) \mid^2 \left(\binom{2^{n-2}}{2^{n-3}} - \mid A_{n-2}(1,4) \mid\right)$.

Let $Y_n^\alpha = T_n^\alpha \cup P_n^\alpha \cup V_n^\alpha \cup Q_n^{1\alpha} \cup R_n^{1\alpha}$. Note that $Y_n^\alpha \subset A_n(1,4)$.

Theorem 6.5.3 *For* $n \geq 4$, $\mid Y_n^\alpha \mid = \mid A_{n-1}(1,4) \mid^2 + 2 \sum_{i=0}^{2^{n-3}-1} C_{n-1}^2(2i) +$
$\left(\binom{2^{n-2}}{2^{n-3}} - \mid A_{n-2}(1,4) \mid\right) (1 + 4 \mid A_{n-2}(1,4) \mid^2)$.

Proof : The proof follows from Proposition 6.5.1 and Proposition 6.5.2. ∎

Here we have discussed several sufficient and necessary conditions for constructing correlation immune functions. We have also considered some other cryptographic criteria such as balancedness, nonlinearity and nondegeneracy. The proof techniques used here are mostly constructive and they provide correlation immune Boolean functions with other important cryptographic properties. Our work clearly highlights that the combinatorial structure of correlation immune Boolean functions is complicated.

We once again mention that the motivation of this chapter is not to present enumerative implications of correlation immune Boolean functins. An asymptotic expression on the cardinality of correlation immune Boolean functions has already been presented by Denisov [28] and all other existing enumeration results are much less sharper with respect to [28]. However, the technique used in [28] is probabilistic. Exact enumeration of correlation immune functions using constructive methods seems to be a very hard problem.

Chapter 7

Design & Implementation of Resilient Boolean Functions

We here present a construction method for cryptographically significant Boolean functions used in stream cipher systems. Siegenthaler proved that an n input 1 output, m-resilient (balanced mth order correlation immune) Boolean function with algebraic degree d satisfies the inequality $m + d \leq n - 1$. We provide a new construction method using a small set of recursive operations for a large class of highly nonlinear, resilient Boolean functions optimizing Siegenthaler's inequality $m + d = n - 1$. Comparisons to previous constructions show that better nonlinearity can be obtained by our method.

Also an efficient hardware implementation strategy for this class of Boolean functions on large number of input variables is presented here. We provide a special representation for such functions so that they can be implemented in VLSI with low cost pipelined architecture. Moreover, the architecture is programmable and can be dynamically reconfigured to compute different functions of the class.

7.1 Introduction

In stream cipher cryptography, the message is considered to be a stream of bits. The cipher is obtained by bitwise XORing (addition over GF(2)) the message with a sequence of bits called the key stream. In most common models of stream ciphers the key stream is produced by using a Boolean function to combine the output sequences of several Linear Feedback Shift Registers (LFSRs). If the combining Boolean function is not properly chosen, then the system becomes susceptible to several kinds of cryptanalytic attacks. An important class of *divide-and-conquer* attacks on such systems was proposed by Siegenthaler [110]. Moreover,

Siegenthaler [109] himself introduced a class of Boolean functions, the set of correlation immune functions, which can resist such attacks. However, it is not sufficient to use functions with only correlation immunity, since certain types of correlation immune functions are susceptible to other kinds of attacks. For example, it is well known that the linear functions are correlation immune but not suitable for use in cryptography. Thus we consider two important cryptographic properties for Boolean functions. The algebráic degree is the degree of the algebraic normal form of a Boolean function. Having a high algebraic degree ensures a high linear complexity of the produced key stream and hence better immunity against the Berlekamp Massey shift register synthesis algorithm [69]. A second measure, nonlinearity, is the distance from the set of affine functions. A high value of this parameter ensures that the best affine approximation [31] attack will fail. Siegenthaler in [109] proved a fundamental inequality relating the number of variables n, order of correlation immunity m and algebraic degree d of a Boolean function: $m+d \leq n$. Moreover, if the function is balanced then $m+d \leq n-1$. Also, a balanced mth order correlation immune function is said to be m-resilient. Since it is natural to use balanced functions in stream cipher systems we concentrate only on resilient functions. A resilient Boolean function is said to be optimized if $m+d = n-1$. Here we provide construction methods for optimized functions having high nonlinearities. The functions are built using a small set of recursive operations and hence functions on large number of variables are easy to implement using nominal hardware.

We use concatenation techniques and introduce generic construction functions (see Definition 7.2.2), which recursively build a correlation immune function of $(n+1)$ variables from two correlation immune functions of n variables. We initially start with bent functions which are modified a little to get optimized algebraic degree. A sequence of such constructors is applied to build correlation immune functions of desired orders from non correlation immune balanced Boolean functions with high nonlinearity. The degree of the resulting function is same as that of the initial function. The method can easily be extended to design functions with moderate to large number of input variables using a special representation of the constructed Boolean functions (see Definition 7.6.1). The actual trade-off between nonlinearity and correlation immunity is explicit (see Theorem 7.6.2). Moreover, we use the functions constructed in Chapter 4 to improve the nonlinearity. The results provide a lower bound on the nonlinearity of a function optimized with respect to Siegenthaler's inequality [109].

Both our technique as well as the technique of [105] can be used to construct highly nonlinear, balanced, n variable, mth order correlation immune (m resilient) functions having algebraic degree $n-m-1$. We show that for all m such that $m+2\log_2(m+3)+3 < n$, the lower bound on nonlinearity obtained by our method for **optimized functions** is better than that of [105]. Thus as n increases, for almost all m, we obtain a better lower bound on nonlinearity. Conversely, if we fix an m, then there exists an N, such that for all $n \geq N$, the lower bound on nonlinearity obtained by our method is better. Moreover, we can use the 1-resilient functions provided in Chapter 4 as initial functions to get improved nonlinearity.

As examples, using our techniques one can construct

1. 10 variable balanced functions with degree 8, order of correlation immunity 1 and nonlinearity 480 and

2. 50 variable balanced functions with degree 20, order of correlation immunity 29 and nonlinearity $2^{49} - 2^{39} + 104 \times 2^{29}$.

Moreover, there are widely different functions in the constructed class (see Example 7.8.1 in Section 7.8).

Current stream cipher systems use a nonlinear Boolean function to combine the output of a small number (8 to 10) of LFSRs to produce a keystream which is used to encrypt the data. With the present speed of computers, using Boolean functions of a small number of input variables (and hence a small number of LFSRs) is sufficient. However, considering the extreme pace in which the present day hardware is advancing, within a few years the known attacks [110, 73, 31, 52, 51, 84] may perform significantly well against the stream cipher systems which use functions on a small number of input variables. Thus, it is important to implement cryptographically strong functions on large number of input variables at a nominal hardware cost. We use the pipelined architecture for this.

Linear pipeline processor is cascade of processing stages which are linearly connected to execute a fixed task over a stream of data flowing from one end to another [47]. The pipelining concept is used in various kinds of hardware design to achieve high processing speed. Recently pipelining architecture has been proposed for MD4 message digest algorithm in Digital Signature [108], LMS algorithm for real time transversal adaptive filtering [94], digital signal processing [56, 72, 29, 85], high performance digital circuit [25, 53] etc.

We here exploit the pipelining technique for increasing speed which also provide a low cost cascadable architecture. Using a special representation (see Definition 7.6.1) a Boolean function f of n $(= m + k + 1)$ variables is built from a Boolean function h of k variables. The value of k is low enough so that h can be easily implemented (either as a lookup table or by a combinational circuit). Given the value of $h(X_k, \ldots, X_1)$ we present two algorithms to compute the value of $f(X_n, \ldots, X_{k+1}, X_k, \ldots, X_1)$ in $(m+1)$ steps. In one approach (top down), we start with the variable X_n and descend step by step to the variable X_{k+1}. In the other approach (bottom up), we start from X_{k+1} and move step by step upto the variable X_n. A direct hardware implementation of any one of the algorithms would require $(m + 1)$ clocks to compute $f(X_n, \ldots, X_1)$. However, using a simple and efficient *store and forward* pipelined architecture having $(m + 1)$ stages, the value of $f(X_n, \ldots, X_1)$ is available at each clock after an initial latency period of $(m + 1)$ clocks. At each clock new values of the variables $X_n, \ldots, X_1$ are input to the first stage of the pipeline. Also at each clock the values stored at the ith stage are forwarded to the $(i + 1)$th stage. The operation performed at

each stage is same for all the stages, giving rise to a regular cascadable design. The circuit required for each stage can be implemented either using combinational circuit or by lookup table. We illustrate this by providing a combinational circuit implementation for the top down algorithm and a lookup table implementation for the bottom up algorithm. Another significant advantage of our implementation is that the hardware for one function can be easily reconfigured dynamically to compute another function of this class. Each stage of the pipeline has a 2-bit control unit, a total of $2(m+1)$ control bits in total. Each function $f(X_n, \ldots, X_1)$ in our class can be uniquely identified by these $2(m+1)$ control bits and the function $h(X_k, \ldots, X_1)$. If $h(X_k, \ldots, X_1)$ is implemented as a lookup table, then by suitably programming h and the $2(m+1)$ control bits one can compute any function of our class. Even if h is fixed, a large number of Boolean functions can be realized by manipulating the $2(m+1)$ control bits.

7.2 Preliminaries

We once again mention some of the notations which we have discussed earlier. Note that the Boolean functions $f(X_1, \ldots, X_n)$ and $f(X_n, \ldots, X_1)$ are same. We sometime use the second kind to keep parity with the truth table representation of the function where the variable X_n is at the leftmost input column and X_1 is the rightmost input column. In the truth table the column corresponding to an input variable X_j occurs to the left of the column corresponding to the input variable X_i, if $j > i$. The first half of the string f is denoted as f^u and the second half is denoted as f^l. If $f \in \Omega_n$, then $f^u, f^l \in \Omega_{n-1}$ and are given by $f^u(X_{n-1}, \ldots, X_1) = f(0, X_{n-1}, \ldots, X_1)$ and $f^l(X_{n-1}, \ldots, X_1) = f(1, X_{n-1}, \ldots, X_1)$. Note that a function $f \in \Omega_n$ may be a non degenerate function of i variables for $i < n$. The reverse of the string S is denoted by S^r. The bitwise complement of a string S is denoted as S^c. If f is a Boolean function, then f^r, the function obtained by reversing the output column of the truth table is given by $f^r(X_n, \ldots, X_1) = f(1 \oplus X_n, \ldots, 1 \oplus X_1)$, where $\oplus$ denotes the XOR operation. Similarly, the function f^c obtained by complementing each bit of the output column of f is given by $f^c(X_n, \ldots, X_1) = 1 \oplus f(X_n, \ldots, X_1)$. We use the convention that $f^{rc} = (f^c)^r = (f^r)^c$.

The set of affine functions of n variables is denoted as $L(n)$. Note that $L(n) = \{H \mid H = hh \text{ or } hh^c, h \in L(n-1)\}$. Let $h \in L(n)$ be a non degenerate function of m $(1 \leq m \leq n)$ variables. If m is even then $h^r = h$ else if m is odd, $h^r = h^c$. The linear function $h \in L(n)$ is degenerate if $m < n$. A high nonlinearity ensures that the best affine approximation cryptanalytic attack will fail. (See [31] for a description of this method). It is well known [95] that for even n, the maximum nonlinearity achievable by a Boolean function is $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$. Such functions are called bent functions and their combinatorial properties have been studied [17, 31, 95]. For odd n, the corresponding class of functions have not

been characterized. Moreover, bent functions are known to be unbalanced and are not correlation immune. Meier and Staffelbach [74] have described a procedure to construct balanced nonlinear functions from bent functions. So if one is looking for functions which optimize Siegenthaler's inequality, one cannot hope to attain the maximum value of $nl(f)$. Another important criterion is algebraic degree, since it determines the linear complexity of the output sequence of the function (see [31]). The relationship of algebraic degree to the order of correlation immunity was studied in [109, 42]. We repeat the following definition of correlation immunity.

Definition 7.2.1 *A function $f(X_n, X_{n-1}, \ldots, X_1)$ is said to be mth $(1 \leq m \leq n-1)$ order correlation immune if $wd(f,h) = 0$ where $h \in L(n)$ and h is a non degenerate function of i variables with $1 \leq i \leq m$. Moreover, if f is balanced then f is called m-resilient.*

From this definition it is clear that *if a function is mth order correlation immune, then it is kth order correlation immune for* $1 \leq k \leq m$. We define,

1. $\Theta_n(m) = \{f \in \Omega_n \mid f$ is correlation immune of order m but not correlation immune of order $m+1\}$.
2. $\Lambda_n(m) = \bigcup_{m \leq k \leq n-1} \Theta_n(k)$, is the set of all correlation immune functions of order m or more.
3. A function is called *correlation immune* if it is at least correlation immune of order one. Also $A_n = \Lambda_n(1)$, is the set of all correlation immune functions of n variables.

Let f be a balanced function of degree d, and $f \in \Theta_n(m)$. Then f is optimized with respect to balancedness, degree and order of correlation immunity if $m+d = n-1$. In Theorem 7.6.2 we describe methods to construct such optimized functions with sufficiently large values of $nl(f)$. We next define three constructions P, Q, R as follows. These constructions have also been used in Chapter 6 to obtain the currently best known lower bounds on (balanced) correlation immune Boolean functions.

Definition 7.2.2

1. $P : \Omega_{n-1} \times \Omega_{n-1} \to \Omega_n,\ P(f,g) = f^u g^u g^l f^l$.
2. $Q : \Omega_{n-1} \times \Omega_{n-1} \to \Omega_n,\ Q(f,g) = fg = f^u f^l g^u g^l$.
3. $R : \Omega_{n-1} \times \Omega_{n-1} \to \Omega_n,\ R(f,g) = f^u g^u f^l g^l$.

It is important to note that we have used all these three constructions earlier in Chapter 6 for the purpose of enumeration. Later we will use these constructions to recursively build correlation immune functions. The construction Q appears in [109], although in a different form. Note that, the generic construction functions P, Q, R should not be viewed as linear combination of two Boolean functions. As example, if we consider the Boolean function $Q(f, f^r)$, then the nonlinearity of $Q(f, f^r)$ will be twice that of f and the number of terms with highest algebraic degree will increase. We discuss it elaborately Section 7.4. In the next section we highlight why it is important to design functions with high order of resiliency.

7.3 Weakness of Resilient Functions

In stream cipher models, a resilient function is used to combine the output sequences of several LFSR's. If a function is correlation immune of order m, then it is not possible to choose a subset of m LFSRs and simultaneously obtain their output sequences. However, if the function is not correlation immune of order $(m+1)$, then it will be possible to attack some subset of $(m+1)$ LFSRs. Here we explain how this weakness can be used to transfer an "attack on non correlation immune functions" to an "attack on correlation immune functions". This idea has also been coined very briefly in [75]. Before proceeding further we require the following well known result on the combination of output sequence of several LFSRs.

Proposition 7.3.1 *Let $\{X_i^t\}$, $1 \leq i \leq k$, $t = 1, 2, \ldots$, be the bit streams generated by LFSRs S_i with connection polynomial $C_i(x)$ and size a_i. Then the bit stream $\{\bigoplus_{i=1}^{k} X_i^t\}$ generated by adding (over GF(2)) outputs of LFSRs S_i is produced by a composite LFSR with connection polynomial $\prod_{i=1}^{k} C_i(x)$ and size $\sum_{i=1}^{m} a_i$.*

Suppose $f(X_1, \ldots, X_n)$ is an n-variable, m-resilient function used in combining the output sequences of n LFSRs S_i having feedback polynomials $C_i(x)$. Further suppose that the output of f is correlated to two linear functions (in $L(n)$) $l_1 = X_{i_1} \oplus \ldots \oplus X_{i_k}$ and $l_2 = X_{i_1} \oplus \ldots \oplus X_{i_k} \oplus X_{i_{k+1}}$, i.e., $wd(f, l_1) \neq 0$ and $wd(f, l_2) \neq 0$ (note that $k > m$). (We have checked that there exists m-resilient functions such that it is possible to get a pair l_1, l_2). Let S_{i_1} to $S_{i_{k+1}}$ be the LFSRs corresponding to the inputs X_{i_1} to $X_{i_{k+1}}$. Further, let CS_1 and CS_2 be two composite LFSRs constructed using Proposition 7.3.1 having output sequences $\bigoplus_{j=1}^{k} X_{i_j}$ and $\bigoplus_{j=1}^{k+1} X_{i_j}$ respectively. The connection polynomials for CS_1 and CS_2 are respectively $\prod_{j=1}^{k} C_{i_j}$ and $\prod_{j=1}^{k+1} C_{i_j}$. Now we can estimate the output sequences of CS_1 and CS_2 using the known techniques [75, 52, 51]. Then it is easy to find the output sequence of X_{k+1} by XORing the two estimates.

We will shortly describe a more general method, but before that let us briefly discuss the practicability of the scheme. The problem is in estimating the output sequences of CS_1 and

CS_2. The length of a composite LFSR is equal to the sum of the lengths of the individual LFSRs. Thus if the time complexity of an attack to estimate the output sequence of an LFSR grows exponentially with the length, then the situation is hopeless. *However, there are attacks where the time complexity does not grow exponentially with the length [75, 52, 51].* In such a situation the attack becomes feasible. For example, the method in [75] can attack upto 1000-bit LFSRs. Since in practical systems the lengths of the LFSRs are around 50 to 70 and the order of resiliency is around 5 to 10, the above mentioned scheme in conjunction with the attack in [75] can be used to attack such systems. The problem here is that the method of [75] does not work if the weight of the feedback polynomial is more than 4 or 5. Since the feedback polynomial for the composite LFSR is a product of different polynomials, in general its weight will be high enough to make the attack infeasible. However, it is not always true that the weight of the product polynomial is greater than those of the factor polynomials. Recent attacks [52, 51] have shown how the weight restriction on the polynomials can be lifted using coding theory based techniques. However, in these cases it is not clear whether these attacks perform well for high length LFSRs. *If the current attacks can be sharpened to attack high length (around 1000) LFSRs with feedback polynomials of reasonable weight, then current systems will have to be substantially modified to ensure security.* In particular, one should choose the order of resiliency m and lengths of the polynomials l in a manner such that $m \times l >> 1000$. Moreover, the order of resiliency m should be small compared to the number of variables, as otherwise the amount of correlation to linear functions of the form l_1 and l_2 described above will be higher on an average. This will make the attacks perform even better.

Next we describe a sharper method to attack each individual input of the combining Boolean function. Given an m-resilient function $f \in \Omega_n$, we define $T_f = \{\overline{\omega} \in \{0,1\}^n \mid F(\overline{\omega}) \neq 0\}$, where F is the Walsh transform of f. Note that, since f is m-resilient, $wt(\overline{\omega}) \geq m+1$. Let $\overline{\omega} = (\omega_1, \ldots, \omega_n)$. Then for $\overline{\omega} \in T_f$, $Prob(f = \bigoplus_{i=1}^{i=n} \omega_i X_i) \neq \frac{1}{2}$. Consider that there exists n linearly independent vectors in T_f. For each such vector $(\omega_1, \ldots, \omega_n)$, it is possible to reconstruct the bit stream corresponding to $\bigoplus_{i=1}^{i=n} \omega_i X_i$. Since the n vectors chosen from T_f are linearly independent, using a linear transformation one can construct n individual streams corresponding to $X_1, \ldots, X_n$. We have checked that there exists m-resilient functions on n variables with n linearly independent vectors in T_f.

Thus, to construct a safe stream cipher system, it is necessary to have functions with high order of resiliency. We have also discussed earlier that high algebraic degree and high nonlinearity are also important. This is only possible if we have simple implementation method of Boolean functions on large number of input variables which we discuss in this chapter.

7.4 Nonlinearity, Algebraic Degree and Balancedness

We provide a few technical results in this section related to nonlinearity, algebraic degree and balancedness.

Theorem 7.4.1 *Let $f, g \in \Omega_{n-1}$ and $F = \Psi(f, g)$ where $\Psi \in \{P, Q, R\}$. Then $nl(F) \geq nl(f) + nl(g)$. Moreover, if $g = f$, $g = f^c$ or $g = f^r$, then $nl(F) = nl(f) + nl(g) = 2nl(f)$.*

Next we state without proof the following result on the degree of the constructed function. The proof consists in checking the different cases.

Theorem 7.4.2 *Let $f \in \Omega_n$ and $F = \Psi(f, f^\tau)$, where $\Psi \in \{P, Q, R\}$ and $\tau \in \{c, r\}$. Then, $deg(F) = deg(f)$.*

The special case of Theorem 7.4.2 with $\Psi = Q$ and $\tau = c$ was mentioned in [33]. The importance of this result lies in the fact that the degree of the constructed function is equal to the degree of the original function. It is known [95] that the degree of bent functions of n variables for $n \geq 4$ is at most $\frac{n}{2}$. We propose the following simple but *powerful* method to improve the degree. Note that, $X_1 \ldots X_n$ means logical AND of X_1 to X_n.

Theorem 7.4.3 *Let $h \in \Omega_n$ be of degree less than n and $f = h \oplus X_1 \ldots X_n$. Then $deg(f) = n$ and $nl(f) \geq nl(h) - 1$. Moreover, if h is a bent function then $nl(f) = nl(h) - 1$.*

Proof : Let, $g \in L(n)$. Then $d(f, g)$ is either $d(h, g) - 1$ or $d(h, g) + 1$. If h is bent, $nl(f) \leq nl(h)$, and so $nl(f) = nl(h) - 1$. ∎

If we start with a bent function $h \in \Omega_8$ and use the above theorem then we can get a function $f \in \Omega_8$ of degree 8 and nonlinearity 119. Using this f, one can get $F = \Psi(f, f^c) \in \Omega_9$, *which is balanced, has degree* 8 *and nonlinearity* 238. Generalizing, we can get balanced functions $F \in \Omega_{2p+1}$ having degree $2p$ and nonlinearity $2^{2p} - 2^p - 2$. It should be noted that for $f \in \Omega_n$, if $deg(f) = n - 1$, then f is not both correlation immune and balanced. Also if $deg(f) = n$, then f is neither correlation immune nor balanced.

Theorem 7.4.3 shows that the degree can be increased significantly with insignificant change in nonlinearity. Moreover, it can be checked that though f in the above theorem has only one term of degree n, the number of terms of degree n in $\Psi(f, f^r) \in \Omega_{n+1}$ is *more than one*. We state one specific result regarding this.

Proposition 7.4.1 *Let $f \in \Omega_n$ with degree n. Then $Q(f, f^r) \in \Omega_{n+1}$ contains n terms of degree n.*

The linear complexity of the output sequence produced by the Boolean function depends on the algebraic normal form of the function and the lengths of the input LFSRs [97, 31]. Having more terms of degree n ensures that the linear complexity of the output sequence is higher. See Example 7.8.1 in the last section for further illustration regarding the number of high degree terms. Proper use of this technique will ensure that the functions designed using Construction 1 (see later), will have this property. This has direct implication towards the stability of the generated sequence [31]. We would like to point out that this phenomenon does not hold for the construction $Q(f, f^c)$ given in [109, 33]. Next, we list a few simple results on balancedness.

Proposition 7.4.2 *(a) A function of the form ff^c is balanced. (b) If f is a balanced function then both f^r and f^c are balanced. (c) Let $f, g \in \Omega_n$ be two balanced functions, and $F = \Psi(f, g)$, where $\Psi \in \{P, Q, R\}$. Then F is also balanced.*

7.5 Correlation Immunity

Here we provide generalized construction methods for correlation immune functions. First we state the following two results which have been proved in different forms in [80, 12] and [109] respectively.

Proposition 7.5.1 *Let $h \in \Omega_n$. Then $Q(h, h^r) = hh^r \in A_{n+1}$.*

Proposition 7.5.2 *Let $f \in \Omega_n$. Then $Q(f, f^c) = ff^c \in A_{n+1}$ iff f is balanced.*

Next we state without proof the following basic result.

Lemma 7.5.1 *Let $f \in \Lambda_n(m)$ (respectively $\Theta_n(m)$). Then $f^r, f^c \in \Lambda_n(m)$ (respectively $\Theta_n(m)$).*

In [109, Section IV] Siegenthaler proposed a construction of $F \in \Lambda_{n+1}(m)$ from $f, g \in \Lambda_n(m)$ as follows.

Theorem 7.5.1 *([109]) If $Z_1 = f_1(X_1, X_2, \ldots, X_n)$ and $Z_2 = f_2(X_1, X_2, \ldots, X_n)$ are mth-order correlation immune functions of n binary variables such that $Prob(Z_1 = 1) = Prob(Z_2 = 1)$, then the binary-valued function f of $n+1$ random variables defined by the GF(2) expression*
$f(X_1, X_2, \ldots, X_{n+1}) = X_{n+1} f_1(X_1, X_2, \ldots, X_n) + (X_{n+1} + 1) f_2(X_1, X_2, \ldots, X_n)$ is also mth order correlation immune.

The condition $Prob(Z_1 = 1) = Prob(Z_2 = 1)$ is equivalent to the condition $wt(f_1) = wt(f_2)$. Note that the construction in the above theorem corresponds to our construction Q. We further generalize the construction to include P, R also.

Lemma 7.5.2 *Let $f, g \in \Lambda_n(m)$ and F be of the form $F = P(f, g) = f^u g^u g^l f^l$. If (a) $m = 1$ or (b) $m > 1$ and $wt(f) = wt(g)$, then $F \in \Lambda_{n+1}(m)$.*

Proof : Let f, g be functions of $\{X_1, X_2, \ldots, X_n\}$ and F be a function of $\{X_1, X_2, \ldots, X_{n+1}\}$. We use the characterization of correlation immunity given in Definition 7.2.1. Let us consider any linear/affine function $H \in L(n+1)$, where H is a non degenerate function of k variables $(1 \leq k \leq m)$.

Now we will have four cases.

1. If H contains k variables from $\{X_1, X_2, \ldots, X_{n-1}\}$ then H is of the form $hhhh$. Now, $wd(F, H) = wd(f^u g^u g^l f^l, hhhh) = wd(f, hh) + wd(g, hh) = 0$, as f, g are mth order correlation immune.

2. If H contains X_n and the remaining $k-1$ variables from $\{X_1, X_2, \ldots, X_{n-1}\}$ then H is of the form $hh^c hh^c$. Then, $wd(F, H) = wd(f^u g^u g^l f^l, hh^c hh^c) = wd(f, hh^c) + wd(g, h^c h) = 0$.

3. If H contains X_{n+1} and the remaining $k-1$ variables from $\{X_1, X_2, \ldots, X_{n-1}\}$ then H is of the form $hhh^c h^c$.
   Now, $wd(F, H) = wd(f^u g^u g^l f^l, hhh^c h^c) = wd(f, hh^c) + wd(g, hh^c) = 0$.

4. If H contains X_n, X_{n+1} and the remaining $k-2$ variables from $\{X_1, X_2, \ldots, X_{n-1}\}$ then H is of the form $hh^c h^c h$. Now two cases arise.
   (a) If $k-2 > 0$, then $wd(F, H) = wd(f^u g^u g^l f^l, hh^c h^c h) = wd(f, hh) + wd(g, h^c h^c) = 0$.
   (b) If $k-2 = 0$, then H is of the form $0^{n-1}1^{n-1}1^{n-1}0^{n-1}$ and hence,
   $wd(F, H) = wd(f^u g^u g^l f^l, 0^{n-1}1^{n-1}1^{n-1}0^{n-1}) = wd(f, 0^n) + wd(g, 1^n) = 0$, if $wt(f) = wt(g)$. Note that the weight condition is not required if $m = 1$.

Hence by Definition 7.2.1, F is mth order correlation immune. ∎

The case for the construction R is similar. Hence we get,

Theorem 7.5.2 *Let $f, g \in \Lambda_n(m)$, with $wt(f) = wt(g)$ and $F = \Psi(f, g)$, where $\Psi \in \{P, Q, R\}$. Then $F \in \Lambda_{n+1}(m)$.*

In [109] only a construction with two correlation immune functions f, g of same order was considered. However, if the correlation immunity of f, g are of different orders then we get the following result.

Theorem 7.5.3 *Let $f \in \Theta_n(m_1)$ and $g \in \Lambda_n(m_2)$ with $m_1 < m_2$. Then $F \in \Theta_{n+1}(m_1)$ if (a) $\Psi = P$ and $m_1 = 1$ or (b) $\Psi = P, Q$ or R and $wt(f) = wt(g)$.*

Proof : The proof that F belongs to $\Lambda_{n+1}(m_1)$ is similar to the above theorem. It can be checked that if $m_1 = 1$ then the weight condition $wt(f) = wt(g)$ is not required for P. To see that $F \in \Theta_{n+1}(m_1)$, note that there exists a function $h \in L(n)$, which is non degenerate of $(m_1 + 1)$ variables such that $wd(f, h) \neq 0$ but $wd(g, h) = 0$. Depending on Ψ we can use this h to build a linear function $H \in L(n+1)$ which is non degenerate of $(m_1 + 1)$ variables such that $wd(F, H) \neq 0$. Hence F is not correlation immune of order $(m_1 + 1)$. ∎

Next we consider construction of $(m+1)$th order correlation immune function from mth order correlation immune functions.

Proposition 7.5.3 *Let f be an n variable balanced function with mth order correlation immunity. Then $F = Q(f, f^c) = ff^c$ is an $(n+1)$ variable function with $(m+1)$th order correlation immunity.*

In a different form, one side of this was first observed in [109] and later in [12]. This is the basic technique of construction used in [33]. We show that the same result can be achieved using R also.

Theorem 7.5.4 *Let $f \in \Theta_n(m)$ and $F = \Psi(f, f^c)$ where $\Psi \in \{Q, R\}$. Then $F \in \Theta_{n+1}(m+1)$ iff f is balanced. Moreover, F is balanced.*

Proof : We prove this theorem for $\Psi = R$, the other case being similar. Let us consider any linear/affine function $H \in L(n+1)$ which is a non degenerate function of k variables $(1 \leq k \leq m+1)$. For $(1 \leq k \leq m)$ the proof that $wd(F, H) = 0$ is similar to that of Lemma 7.5.2. If H is a non degenerate function of $(m+1)$ variables then H can be of the forms $hhhh$, hhh^ch^c, hh^chh^c and hh^ch^ch. Let H be of the form hh^chh^c, where $h \in L(n-1)$ is non degenerate of m variables. So, $wd(R(f, f^c), H) = wd(R(f, f^c), hh^chh^c) = wd(f, hh) + wd(f^c, h^ch^c) = 2wd(f, hh) = 0$ as $f \in \Theta_n(m)$ and hh is a non degenerate function of m variables. It can be checked that for the other cases also $wd(R(f, f^c), H) = 0$. This shows that $F \in \Lambda_{n+1}(m+1)$.

The resulting $R(f, f^c)$ will not be in $\Lambda_{n+1}(m+2)$. We show a function $H \in L(n+1)$ which is a non degenerate function of $(m+2)$ variables, such that $wd(R(f, f^c), H) \neq 0$. Since f is not correlation immune of order $(m+1)$, there exists a non degenerate function $h_1 \in L(n)$ of $(m+1)$ variables such that $wd(f, h_1) \neq 0$. Now two cases arise.
Case 1: h_1 is of the form hh, where $h \in L(n-1)$. Then h is nondegenerate of $(m+1)$ variables and let $H \in L(n+1)$ be of the form hh^chh^c. Then, $wd(R(f, f^c), H) = wd(f^u(f^u)^cf^l(f^l)^c, hh^chh^c) = wd(f, hh) + wd(f^c, h^ch^c) = 2wd(f, hh) \neq 0$.

Case 2: h_1 is of the form hh^c, where $h \in L(n-1)$. In this case h is non degenerate of m variables and take $H \in L(n+1)$ to be of the form hh^ch^ch. Now, $wd(R(f,f^c),H) = wd(f^u(f^u)^cf^l(f^l)^c, hh^ch^ch) = wd(f,hh^c) + wd(f^c,h^ch) = 2wd(f,hh^c) \neq 0$. ■

The above result does not in general hold for the construction P. If h_1 in the above proof is of the form hh^c, and we choose H to be of the form hh^chh^c, which is non degenerate of $(m+1)$ variables, then $wd(P(f,f^c),H) = wd(f^u(f^u)^c(f^l)^cf^l, hh^chh^c) =$
$wd(f,hh^c) + wd(f^c,h^ch) = 2wd(f,hh^c) \neq 0$. However, the following result holds.

Lemma 7.5.3 *Let $f \in \Omega_n - A_n$ be such that $wt(f^u) = wt(f^l)$. Then $P(f,f^c) \in A_{n+1}$ and is balanced.*

If f is a correlation immune function of even order then we can use f^r instead of f^c in Theorem 7.5.4.

Theorem 7.5.5 *Let $f \in \Theta_n(m)$ and $\Psi \in \{Q,R\}$.*
1. Let $F = \Psi(f,f^r)$. Then, $F \in \Theta_{n+1}(m+1)$ iff m is even. Moreover, F is balanced iff f is balanced.
2. Let $F = \Psi(f,(f^r)^c)$. Then, $F \in \Theta_{n+1}(m+1)$ iff m is odd. Moreover, F is balanced.

Proof : We only prove (1) for $\Psi = R$. We show that if $H \in L(n+1)$, and H is a non degenerate function of k $(1 \leq k \leq m+1)$ variables, then $wd(R(f,f^r),H) = 0$. The case where $1 \leq k \leq m$ is similar to Lemma 7.5.2. Now for $k = m+1$ four cases arise.

1. H is of the form $hhhh$. Then $h \in L(n-1)$ and h is a non degenerate function of $(m+1)$ variables. Since m is even, $(m+1)$ is odd and so $h^r = h^c$. Therefore, $wd(R(f,f^r),hhhh)) = wd(f,hh)+wd(f^r,hh) = wd(f,hh)+wd(f,h^rh^r) = wd(f,hh)+$ $wd(f,h^ch^c) = wd(f,hh) - wd(f,hh) = 0$.

2. H is of the form hh^chh^c. Then hh^c is a non degenerate function of $(m+1)$ variables and hence h is a non degenerate function of m variables. Therefore,
$wd(R(f,f^r),hh^chh^c) = wd(f,hh) + wd(f^r,h^ch^c) = 0+0 = 0$ as $f, f^r \in \Theta_n(m)$.

3. H is of the form hhh^ch^c. Then hh is a non degenerate function of m variables and hence hh^c is a non degenerate function of $(m+1)$ variables. Therefore,
$wd(R(f,f^r),hhh^ch^c) = wd(f,hh^c) + wd(f^r,hh^c) = wd(f,hh^c) + wd(f,(hh^c)^r)$
$= wd(f,hh^c) - wd(f,hh^c) = 0$.

4. H is of the hh^ch^ch. Then h is a non degenerate function of $(m-1)$ variables and so hh^c is a non degenerate function of m variables.
Hence, $wd(R(f,f^r),hh^ch^ch) = wd(f,hh^c) + wd(f^r,h^ch) = 0+0 = 0$.

Hence $wd(R(f, f^r), H) = 0$ and so $R(f, f^r) \in \Lambda_{n+1}(m+1)$. The proof that $R(f, f^r) \notin \Lambda_{n+1}(m+2)$ is similar to Theorem 7.5.4. Thus if m is even, $F \in \Theta_{n+1}(m+1)$.

If m is odd, we show that $F \notin \Theta_{n+1}(m+1)$. Since $f \in \Theta_n(m)$, there exists a function $h_1 \in L(n)$, which is nondegenerate on $m+1$ variables such that $wd(f, h_1) \neq 0$. Now two cases can arise.

1. Consider h_1 is of the form hh, where $h \in L(n-1)$ and h is a non degenerate function of $(m+1)$ variables. Now consider the linear function $H \in L(n+1)$ where $H = hhhh$. Thus H is nondegenerate on $(m+1)$ variables. Now $wd(R(f, f^r), hhhh)) = wd(f, hh) + wd(f^r, hh) = wd(f, hh) + wd(f^r, h^r h^r) = 2wd(f, hh) \neq 0$, since $h^r = h$ as h is nondegenerate on even number of variables.

2. Consider h_1 is of the form hh^c, where $h \in L(n-1)$ and h is a non degenerate function of m variables. Now consider the linear function $H \in L(n+1)$ where $H = hh^c hh^c$. Thus H is nondegenerate on $(m+1)$ variables. Now $wd(R(f, f^r), hh^c hh^c)) = wd(f, hh) + wd(f^r, h^c h^c) = wd(f, hh) + wd(f^r, h^r h^r) = 2wd(f, hh) \neq 0$, since $h^r = h^c$ as h is nondegenerate on odd number of variables.

This proves the other side. ■

Camion et al [12] had earlier observed both (1) and (2) of the above theorem for $\Psi = Q$.

Remark 7.5.1 *In Theorem 7.5.4 and Theorem 7.5.5 we can obtain a weaker result by replacing* $\Theta_n(m)$ *and* $\Theta_{n+1}(m+1)$ *by* $\Lambda_n(m)$ *and* $\Lambda_{n+1}(m+1)$ *respectively.*

We also have the following result which is similar to Lemma 7.5.3.

Lemma 7.5.4 *Let* $f \in \Omega_n - A_n$ *be such that* $wt(f^u) = wt(f^l)$. *Then* $P(f, f^r) \in A_{n+1}$.

Consider $f(X_n, \ldots, X_1), g(X_n, \ldots, X_1) \in \Omega_n$. Now we construct $F_{n+1}(X_{n+1}, \ldots, X_1) = (1 \oplus X_{n+1}) f(X_n, \ldots, X_1) \oplus X_{n+1} g(X_n, \ldots, X_1)$. Note that $F_{n+1} = Q(f, g)$. Similarly, in general we can consider the case $F_i(X_{n+1}, \ldots, X_1) = (1 \oplus X_i) f(X_{n+1}, \ldots, X_{i+1}, X_{i-1}, \ldots, X_1) \oplus X_i g(X_{n+1}, \ldots, X_{i+1}, X_{i-1}, \ldots, X_1)$. Note that the functions F_{n+1} and F_i have the same properties, since F_i is obtained from F_{n+1} by a permutation of the input variables. Note that for implementation in hardware we will only consider the operators Q, R, where the variables X_{n+1} or X_n will be taken out. We do not consider the general case taking any variable X_i out since for hardware implementation it is simpler to use only Q, R.

We will be using the results of Section 7.4 and Section 7.5 to design cryptographically strong Boolean functions in the next section.

7.6 Generalized Construction

Here we describe a recursive procedure to design *highly nonlinear* Boolean functions which *optimizes balancedness, degree* and *order of correlation immunity.* Such functions are ideally suited for stream cipher applications since they can resist all known types of attacks. First we require the following definition.

Definition 7.6.1 *Let $(S_i)_{1 \leq i \leq q}$ be a finite sequence, where,*
$S_i \in \{Q, R\} \times \{c, r, rc\}$. Given a function $h \in \Omega_k$ and a sequence S_i of length q we define a function $F \in \Omega_{q+k}$ as follows.
$F_0 = h$ and $F_i = \Psi_i(F_{i-1}, F_{i-1}^{\tau_i})$
where $S_i = (\Psi_i, \tau_i)$, for $i \geq 1$, and $F = F_q$. We say that F is represented by $(h, S_1, \ldots, S_q)$ and the length of the representation is q.

First we observe that given a function $h \in \Omega_k$ it is easy to design a linear time (on the number of inputs to the function) algorithm that generates a function $F \in \Omega_{q+k}$ represented by $(h, S_1, \ldots, S_q)$. Though the size of the function F may be large, we need not store the whole truth table for F. Using the representation of F, the storage space required is not much larger than h. The penalty is that we require an algorithm to calculate the output of F. This can be done in $O(q)$ time (specifically, q clocks in hardware circuit) if h is implemented as a truth table. However, very low cost pipelined circuit (using flip flops) can be developed which produces a output at each clock pulse after an initial latency period of q clocks. Both the hardware and the algorithm are interesting which we explain in Section 7.9. Now we state some important properties of functions constructed by the above procedure.

Theorem 7.6.1 *Let $h \in \Omega_k$ and $F \in \Omega_{m+k+1}$ be represented by $(h, S_1, \ldots, S_{m+1})$ where $S_i = (\Psi_i, \tau_i)$, $\tau_{2i+1} \in \{c, rc\}$ and $\tau_{2i+2} \in \{c, r\}$ for $i \geq 0$. Then F is balanced and (1) $nl(F) = 2^{m+1} nl(h)$ (2) $deg(F) = deg(h)$, (3) If $m \geq 1$, then $F \in \Lambda_{m+k+1}(m)$. Moreover, if degree of h is k, then $F \in \Theta_{m+k+1}(m)$.*

Proof : (1) Follows from Theorem 7.4.1. (2) Follows from Theorem 7.4.2. (3) Follows from Theorem 7.5.4, Theorem 7.5.5 and Remark 7.5.1. Moreover, if degree of h is k, then F can not be correlation immune of order $m+1$ due to the Siegenthaler's inequality. ■

Note that there are four possible options of S_i for $i > 0$. Moreover, the construction P can also be used in the first step S_1, since the purpose of the first step is to attain balancedness. This generalizes the construction method of [33, Section 5], which uses the sequence $S_i = (Q, c)$ for all $i \geq 1$.

Corollary 7.6.1 *Let $h \in \Omega_k$ be balanced and $F \in \Omega_{m+k}$ be represented by $(h, S_1, \ldots, S_m)$, where $S_i = (\Psi_i, \tau_i)$, $\tau_{2i+1} \in \{c, r\}$ and $\tau_{2i+2} \in \{c, rc\}$*

for $i \geq 0$. *Then* F *is in* $\Lambda_{m+k}(m)$. *Moreover, if degree of* h *is* $(k-1)$, *the maximum degree attained for a balanced function, then* $F \in \Theta_{m+k}(m)$.

It is important to realize that there are different trade-offs involved among the parameters, algebraic degree $deg(.)$, order of correlation immunity m, nonlinearity $nl(.)$, balancedness and the number of input variables n. The first result from [109], is that for any Boolean function f, $deg(f) + m \leq n$ and for balanced Boolean functions, $deg(f) + m \leq n - 1$. The next result is that the maximum value of nonlinearity for even n is achieved for bent functions and it is known [95] that for $n \geq 4$, the degree of such functions cannot exceed $\frac{n}{2}$. Let us now consider the following construction which provides a good trade-off among the parameters.

Construction 1. *On input* n, m *we provide a method to construct a balanced* n *variable* m*th order correlation immune function with algebraic degree* $k = n - m - 1$. *Let* $h \in \Omega_k$ *of degree* k *be as follows.*
If k *is even, then* h *is formed by adding the term* $X_1 \ldots X_k$ *(logical AND of* X_1 *to* X_k*) to a bent function* g *of* k *variables. If* k *is odd then* h *is formed by adding the term* $X_1 \ldots X_k$ *to a function* g *of* k *variables, where* g *is formed by concatenating two bent functions of* $(k-1)$ *variables.*
Let $F \in \Omega_n$ *where* $n = m + k + 1$ *and* $m \geq 1$. F *is represented by* $(h, S_1, \ldots, S_{m+1})$ *where* $S_i = (\Psi_i, \tau_i)$, $\Psi_i \in \{Q, R\}$, $\tau_{2i+1} \in \{c, rc\}$ *and* $\tau_{2i+2} \in \{c, r\}$ *for* $i \geq 0$.

It is clear from the above discussion that Construction 1 provides functions which optimize Siegenthaler's inequality. We now find out the exact expression of nonlinearity obtained by the above construction. The result follows from Theorem 7.6.1, Corollary 7.6.1 and the nonlinearity of bent functions.

Theorem 7.6.2 *Consider* $F \in \Theta_n(m)$ *as in Construction 1.*
(1) If $n \not\equiv m \bmod 2$, *then* $nl(F) = 2^{n-1} - 2^{\frac{n+m-1}{2}} - 2^{m+1}$.
(2) If $n \equiv m \bmod 2$ *then* $nl(F) \geq 2^{n-1} - 2^{\frac{n+m}{2}} - 2^{m+1}$.

Proof : (1) We take a bent function g of $k = n-m-1$ variable and $h = g \oplus X_1 X_2 \ldots X_{n-m-1}$. Thus by Theorem 7.4.3, $nl(g) = 2^{n-m-2} - 2^{\frac{n-m-1}{2}-1} - 1$. Then we apply our method of Definition 7.6.1 to get $nl(f) = 2^{m+1}\,(2^{n-m-2} - 2^{\frac{n-m-1}{2}-1} - 1)$.
(2) We take a bent function g_1 of $n - m - 2$ variables. Then we use bent concatenation to get g of $n - m - 1$ variables with nonlinearity $nl(g) = 2^{n-m-2} - 2^{\frac{n-m-2}{2}}$. Now, $h = g \oplus X_1 X_2 \ldots X_{n-m-1}$. Thus, $nl(h) \geq 2^{n-m-2} - 2^{\frac{n-m-2}{2}} - 1$. Hence, $nl(f) \geq 2^{m+1}\,(2^{n-m-2} - 2^{\frac{n-m-2}{2}} - 1)$. ∎

This also shows that by varying the order of correlation immunity, one can adjust the nonlinearity of the optimized functions. However, the result in Theorem 7.6.2 can be im-

proved using the following two results (Theorem 4.3.2 and Theorem 4.3.3) of Chapter 4 which we state once more.

Theorem 7.6.3 *For even $n \geq 4$, it is possible to construct n-variable, 1-resilient functions with degree $n-2$ and nonlinearity $x = 2nla(n-1)$. The value of x is equal to $2^{n-1} - 2^{\frac{n}{2}}$ for $4 \leq n \leq 14$ and is greater than $2^{n-1} - 2^{\frac{n}{2}}$ for $n \geq 16$.*

Theorem 7.6.4 *For odd $n \geq 5$, it is possible to construct 1-resilient, degree $(n-2)$ functions in Ω_n with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$.*

Next we provide an updated construction method.

Construction 2. *On input n, m we provide a method to construct a balanced n variable mth order correlation immune function with algebraic degree $k = n-m-1$. Here we consider $1 \leq m < n-2$.*

Let h be an $(n-m+1)$-variable, 1-resilient, degree $(n-m-1)$ function constructed using Theorem 7.6.3 or Theorem 7.6.4 according as $(n-m+1)$ is even or odd. Use the recursive construction method (similar to Construction 1), $(m-1)$ times to obtain an n-variable, m-resilient, degree $(n-m-1)$ function F with nonlinearity $2^{m-1}nl(h)$. F is represented by $(h, S_1, \ldots, S_{m-1})$ where $S_i = (\Psi_i, \tau_i)$, $\Psi_i \in \{Q, R\}$, $\tau_{2i+1} \in \{c, rc\}$ and $\tau_{2i+2} \in \{c, r\}$ for $i \geq 0$.

Theorem 7.6.5 *The above construction method constructs n-variable, m-resilient, degree $(n-m-1)$ functions with nonlinearity $nl(f)$, where (1) $nl(f) = 2^{n-1} - 2^{\frac{n+m-2}{2}}$ if $n-m+1$ is odd and (2) $nl(f) = 2^m nla(n-m)$ if $n-m+1$ is even. Further for even $n-m+1$, $nl(f) = 2^{n-1} - 2^{\frac{n+m-1}{2}}$ if $4 \leq n-m+1 \leq 14$ and $nl(f) > 2^{n-1} - 2^{\frac{n+m-1}{2}}$ if $n-m+1 \geq 16$.*

Clearly Construction 2 provides better nonlinearity than Construction 1. In particular, it is possible to construct a 50-variable optimized 29-resilient functions with nonlinearity $2^{49} - 2^{39} + 104 \times 2^{29}$ ($2^{49} - 2^{39} - 2^{30}$ using Construction 1).

7.7 Direct Construction

Seberry, Zhang and Zheng [105] provided construction of resilient functions and analysed the algebraic degree, nonlinearity and propagation characteristics of such functions. Here we provide a simpler interpretation of the construction method provided in [105] and show that this also gives simpler proofs for the order of correlation immunity and nonlinearity of the constructed functions. Let $L(n,k)$ be the set of all $f \in L(n)$, which are the sum modulo 2 (XOR) of exactly k variables and $MU(n,k) = L(n,k) \cup L(n,k+1) \cup \ldots \cup L(n,n)$. Also let $ML(n,k) = L(n,1) \cup L(n,2) \cup \ldots \cup L(n,k)$.

Definition 7.7.1 *Let $n = n_1 + n_2$ and choose 2^{n_1} functions $f_0, \ldots, f_{2^{n_1}-1}$ from the set $MU(n_2, m+1)$. Let $f = f_0 \ldots f_{2^{n_1}-1}$, and denote by $\Gamma(n, n_2, m)$ the set of all such functions. Clearly $\Gamma(n, n_2, m) \subseteq \Omega_n$.*

We first state a simple result which is crucial to understand the cryptographic properties of the construction provided by Definition 7.7.1. The proof is a simple consequence of the fact that the XOR of two linear functions is also a linear function.

Proposition 7.7.1 *Let $l_1, l_2 \in L(n)$. (a) If $l_1 = l_2$ then $d(l_1, l_2) = 0$. (b) If $l_1 = l_2^c$ then $d(l_1, l_2) = 2^n$ and (c) If $l_1 \neq l_2$ or l_2^c then $d(l_1, l_2) = 2^{n-1}$. Consequently, the Walsh distances are respectively, 2^n, -2^n and 0.*

The following result is easy to see.

Proposition 7.7.2 *The construction provided in Definition 7.7.1 is same as that given by Equation 5 of [105].*

This proves (using [105, Corollary 8]) that any function in $\Gamma(n, n_2, m)$ is an mth order CI function. Here we provide a much simpler direct proof as follows.

Theorem 7.7.1 $\Gamma(n, n_2, m) \subseteq \Lambda_n(m)$.

Proof : Let $f \in \Gamma(n, n_2, m)$. We show that for any $l \in L(n, k)$, $wd(f, l) = 0$ for all $1 \leq k \leq m$. We write $f = f_0 \ldots f_{2^{n_1}-1}$, where each $f_i \in MU(n_2, m+1)$. It is not difficult to see that l can be written as $l_0 \ldots l_{2^{n_1}-1}$, where each $l_i \in ML(n_2, m)$. Now $wd(f, l) = wd(f_0 \ldots f_{2^{n_1}-1}, l_0 \ldots l_{2^{n_1}-1}) = \sum_{i=0}^{2^{n_1}-1} wd(f_i, l_i) = 0$, using Proposition 7.7.1, since $f_i, l_i \in L(n_2)$ and $f_i \neq l_i$ or l_i^c. ∎

Visualizing the construction as above, it is easy to obtain the nonlinearity as follows.

Theorem 7.7.2 *Let $f \in \Gamma(n, n_2, m)$ be of the form $f_0 \ldots f_{2^{n_1}-1}$, where each $f_i \in MU(n_2, m+1)$. Then $nl(f) \geq 2^{n-1} - t2^{n_2-1}$, where t is the maximum number of times a function h or its complement h^c are together repeated in the construction $f_0 \ldots f_{2^{n_1}-1}$ for some $h \in MU(n_2, m+1)$.*

Proof : Let $l \in L(n)$. We have to show that $d(f, l)$ is at least as large as the given bound. Note that l can be written as $l_0 \ldots l_{2^{n_1}-1}$, where each l_i is either g or g^c for some $g \in L(n_2)$. Then at most t of the l_i's and f_i's can be equal. Using Proposition 7.7.1, it follows $d(l, f) \geq (2^{n_1} - t)2^{n_2-1} = 2^{n-1} - t2^{n_2-1}$. ∎

Theorem 7.7.2 is first proved in [105, Theorem 14]. One can show as in [105, Theorem 12], that the degree of such functions is $n - n_2 + 1$, provided there are at least two functions

g_1, g_2 among the f_i's of Theorem 7.7.2, such that $g_1 \neq g_2$ or g_2^c and there is a variable which occurs in an odd number of these f_i's. Thus maximum degree is attained if $n_2 = m + 2$, in which case the constructed function optimizes Siegenthaler's inequality.

Let us now estimate the nonlinearity of functions constructed using the method of [105], for functions which optimize Siegenthaler's inequality.

Let $\Omega_{k,n} = MU(n, k+1)$. *By nld(n), we denote the lower bound on nonlinearity of n-variable optimized functions achieved by the direct construction of Definition 7.7.1 (see also [105]).* Note that Siegenthaler's inequality is optimized if $n_2 = m + 2$ and in this case, $|\Omega_{m,m+2}| = \binom{m+2}{m+1} + \binom{m+2}{m+2} = m + 3$. Since one has to choose 2^{n_1} functions from $\Omega_{m,m+2}$, the repetition factor t is at least $\lceil \frac{2^{n-m-2}}{m+3} \rceil$ and hence the nonlinearity obtained is
$nld(n) = 2^{n-1} - \lceil \frac{2^{n-m-2}}{m+3} \rceil \, 2^{m+1}$.

The construction method provided in Section 7.6 is a recursive concatenation of highly nonlinear Boolean functions. On the other hand, the construction provided in Definition 7.7.1 is a direct concatenation of linear functions.

7.8 Results and Comparison to Previous Research

First we compare $nld(n)$, the lower bound of nonlinearity in [105], with our method. *For n variable functions, let the lower bound of nonlinearity obtained by Construction 1 be nlr(n).*

1. **When** $n \not\equiv m \bmod 2$.

$nlr(n) = 2^{n-1} - (2^{\frac{n-m-3}{2}} + 1)\, 2^{m+1}$. $nld(n) = 2^{n-1} - \lceil \frac{2^{n-m-2}}{m+3} \rceil \, 2^{m+1}$. So, our method works favourably when
$\lceil \frac{2^{n-m-2}}{m+3} \rceil > 2^{\frac{n-m-3}{2}} + 1.$ (I)
We consider it more conservatively, i.e., we replace $(2^{\frac{n-m-3}{2}} + 1)$ by $2^{\frac{n-m-2}{2}}$. Hence, our method performs better when,
$\lceil \frac{2^{n-m-2}}{m+3} \rceil > 2^{\frac{n-m-2}{2}}$, i.e., $\lceil \frac{2^{n-m-2}}{2^{\log_2(m+3)}} \rceil > 2^{\frac{n-m-2}{2}}$, i.e., $n - m - 2 - \log_2(m+3) > \frac{n-m-2}{2}$, i.e., when,
$n > m + 2\log_2(m+3) + 2.$ (IA)

2. **When** $n \equiv m \bmod 2$.

$nlr(n) = 2^{n-1} - (2^{\frac{n-m-2}{2}} + 1)\, 2^{m+1}$. $nld(n) = 2^{n-1} - \lceil \frac{2^{n-m-2}}{m+3} \rceil \, 2^{m+1}$. So, our method works favourably when $\lceil \frac{2^{n-m-2}}{m+3} \rceil > 2^{\frac{n-m-2}{2}} + 1.$ (II)
We consider it more conservatively, i.e., we replace $(2^{\frac{n-m-2}{2}} + 1)$ by $2^{\frac{n-m-1}{2}}$. Hence, our method performs better when,
$\lceil \frac{2^{n-m-2}}{2^{\log_2(m+3)}} \rceil \geq 2^{\frac{n-m-1}{2}}$, i.e., $\lceil \frac{2^{n-m-2}}{2^{\log_2(m+3)}} \rceil > 2^{\frac{n-m-1}{2}}$, i.e., $n - m - 2 - \log_2(m+3) > \frac{n-m-1}{2}$, i.e., when

n	$nlx(n)$	M	$nld(n)$	A	$nlr(n)$	B
8	112, 5	112, -	96, 6	112, 6	108, 6	112, 6
9	240, 5	232, -	192, 7	236, 7	220, 7	240, 7
10	480, 6	476, -	384, 8	480, 8	476, 8	480, 8
11	992, 7	976, -	768, 9	976, 9	956, 9	992, 9
12	1984, 7	1972, -	1536, 10	-	1980, 10	1984, 10
13	4032, 7	-	3072, 11	-	3964, 11	4032, 11

Table 7.1: Comparing the nonlinearities of resilient functions

$n > m + 2\log_2(m+3) + 3.$ (IIA)

One can look at (I) and (II) in two ways.

1. If we fix a particular value of m, then there is a certain N, such that $nlr(n) > nld(n)$ for all $n \geq N$. For example for $m = 1$, our method performs better for all $n \geq 8$.

2. Similarly, if we fix a value of n, we get an upper bound $M(n)$ on m, such that for all $m \leq M(n)$, we have $nlr(n) > nld(n)$. Moreover, from (IA) and (IIA), it is clear that this upper bound $M(n)$ becomes close to n, as n increases.

This clearly shows that in a majority of cases the functions obtained by our method are better than those obtained using [105]. *It is also clear that Construction 2 provides better nonlinearity than construction 1.*

It should be noted that high nonlinearity can be obtained by the direct construction provided in [105] without optimizing the Siegenthaler's inequality. However, the nonlinearity of this method decreases when the optimization criterion is considered. From [105, Theorem 12, 14], if one does not want to optimize the Siegenthaler's inequality, then the nonlinearity for first order correlation immune functions is (we denote it as $nlx(n)$ for n variable function) $nlx(n) = 2^{n-1} - min_{3 \leq r < n}(\lceil \frac{2^{n-r}}{2^r - r - 1} \rceil 2^{r-1})$ with algebraic degree $n - r + 1$. In the Table 7.1 we compare nonlinearities of first order CI functions. In second, third and fourth columns we respectively provide nonlinearities of nonoptimized $(m + d < n - 1)$ functions constructed using the method of [105], optimized functions constructed using the method of [105] and optimized functions constructed using our recursive method. Each of the entries are **<nonlinearity, algebraic degree>**. Note that, the Equations $(IA), (IIA)$ provide a clear analysis of when the lower bound of nonlinearity in the recursive construction is better than that of [105]. The table only illustrates this point for small values of n. Here, the column A provides the result obtained in [86] which was obtained by randomized heuristic and B provide the result of construction 2. Also M represents the result obtained in [78] using heuristic search, though the algebraic degree of the functions are not mentioned. It should also be noted that the heuristic search techniques proposed in [78, 86] is not suitable

for constructing functions with large number of input variables.

Example 7.8.1 *The class of functions constructed by our recursive method contains significantly different functions. As a simple example for* $n = 10$, *and* $m = 1$, *let* $f_1 = (h, (Q, c), (Q, r))$ *and* $f_2 = (h, (R, c), (R, r))$, *where* $h \in \Omega_8$ *and is modified from bent functions as in Construction 1. As a concrete example,*
$h = \oplus_{i=1}^{4} X_i Y_i \oplus X_1 X_2 X_3 X_4 \oplus X_1 \ldots X_4 Y_1 \ldots Y_4$. *Then both* f_1, f_2 *contains 8 terms of degree 8. Moreover, the function* $f_1 \oplus f_2$ *is nondegenerate and contains 14 terms of degree 8.*

Next we compare the performance of our construction with [33]. In [33, Section 5], balanced $g \in \Omega_9$ with $nl(g) = 224$, correlation immunity of order 2 and degree 6 has been reported. The function is optimized with respect to Siegenthaler's inequality. The function g has the representation $(f, (Q, c), (Q, c))$, where $f \in \Omega_7$ and has degree 6 (only one term) and nonlinearity 56. It was remarked in [33, Example 5] that g is the representative of all such functions obtained which are well-suited for stream cipher application. Using this function f as our initial function, one can construct *more functions* of the form (f, S_1, S_2) as in Corollary 7.6.1, with the same parameters as g above. As an example one can construct a function of the form $(f, (Q, r), (Q, c))$, which contains 6 terms of degree 6. However, it seems difficult to get such good functions f for a larger number of input variables. The particular function $f \in \Omega_7$ reported in [33, Section 5] was obtained by exhaustive search over a particular subset (the idempotents) of Boolean functions. It seems infeasible to carry out such an exhaustive search for functions of larger number of input variables. Using our method from scratch, if one starts with a bent function $f_1 \in \Omega_6$ and apply Construction 1, we get a balanced, second order correlation immune function g_1 with degree 6 and $nl(g_1) = 216$. The direct construction method in [105] provides a lower bound on maximum nonlinearity $2^{9-1} - \lceil \frac{2^{9-2-2}}{2+3} \rceil 2^{2+1} = 200$. Using Theorem 7.6.5 we get

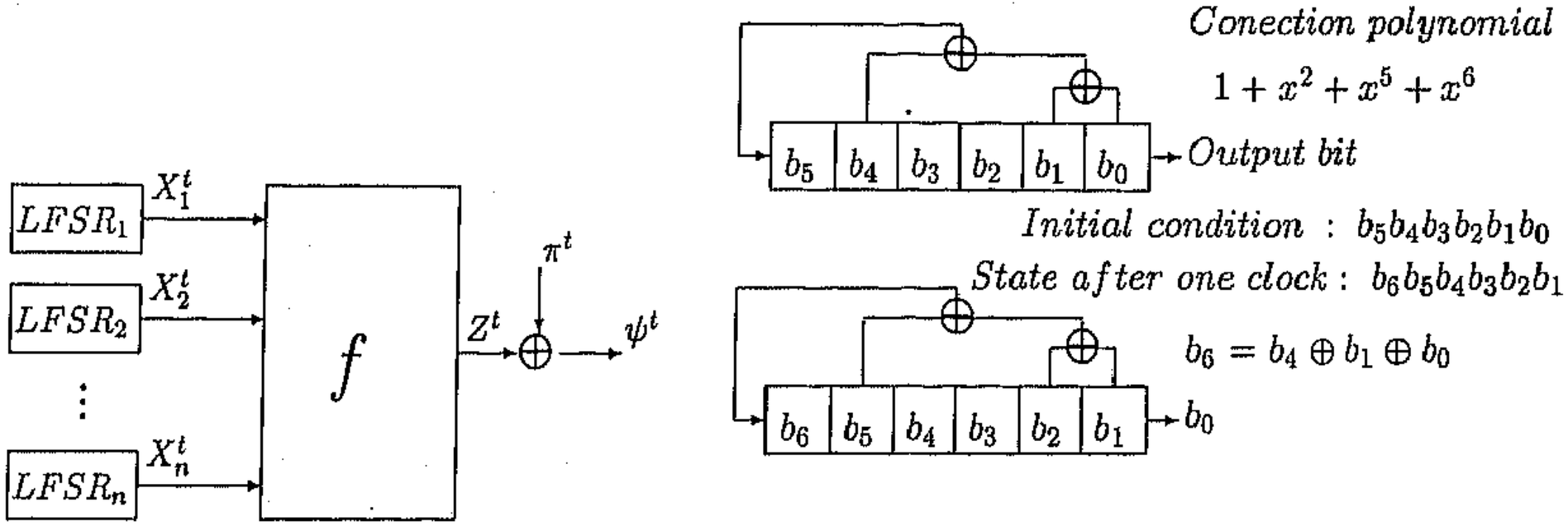
1. 2-resilient, degree 6 functions on 9 variables with nonlinearity 224

2. 3-resilient, degree 5 functions on 9 variables with nonlinearity 224.

Both these functions are optimized and need not use any search technique. Moreover, the method in [33] can not provide the 3-resilient function provided in item 2.

In [86], optimized 1-resilient functions with nonlinearity 112 (for 8 variables) and 480 (for 10 variables) have been obtained. Theorem 7.6.5 constructs optimized 2-resilient functions with nonlinearity 112 (for 8 variables) and 480 (for 10 variables). Also it is clear that the result provided by our methods are much better than what obtained in [78] using heuristic search techniques.

The recursive method proposed here can be used effectively to construct functions with large number of variables. As an example, using Construction 1, if we take a bent function



(a) *LFSR based encryption scheme* (b) *LFSR : One step evolution*

Figure 7.1: Stream Cipher System

$h \in \Omega_{20}$, then modify it to get f as in Theorem 7.4.3 and consider a F represented by a sequence $(f, S_1, \ldots, S_{30})$ satisfying Theorem 7.6.2, then $F \in \Theta_{50}(29)$ with nonlinearity $2^{49} - 2^{39} - 2^{30}$ and $deg(F) = 20$. Moreover, if we use Construction 2, we can start with a 1-resilient function $f \in \Omega_{22}$ with nonlinearity $2^{21} - 2^{11} + 2 \times 104$ (we have provided the construction of balanced 21-variable function with nonlinearity $2^{20} - 2^{10} + 104$ in Chapter 5) and use a sequence $(f, S_1, \ldots, S_{28})$. This will provide a function in $\Theta_{50}(29)$ with nonlinearity $2^{49} - 2^{39} + 104 \times 2^{29}$.

Direct implementation of F using truth table will take 2^{50} bits, which is not feasible to store. However, using the representation of F as $(f, S_1, \ldots, S_{30})$, it is possible to implement F efficiently. Next we describe the efficient pipelining architecture to implement the design in hardware.

7.9 Algorithms and Hardware

In this section we provide algorithms and hardware for resilient functions on large number of input variables. We explain the method corresponding to Construction 1 only. The method is similar for Construction 2.

In LFSR based stream cipher applications the initial conditions of the LFSRs are considered as the secret key. See [76, Page 196] for more details about LFSR. In such a system,

n bits from the n different LFSR's are generated in each clock as n input values to the combining function. On the other hand both the algorithms we propose here need one step for initialization and then $m+1$ steps in loop to generate the output, i.e., $m+2$ clocks in the hardware circuit. Thus, it is important to map the algorithm on a specific hardware architecture so that the output can be available at each clock. We design a store and forward pipelining framework for this purpose which implements the algorithms proposed here. Also, to best of our knowledge, presently no synthesis method is available for such a function with complicated algebraic normal from.

We consider the functions described by Construction 1 and present algorithms and hardware to calculate the output of such a function F on an n bit input $(X_n, \ldots, X_{k+1}, X_k, \ldots, X_1)$. The function F is represented by $(h(X_k, \ldots, X_1), S_1, \ldots, S_{m+1})$, where h is presented as a black box and can be implemented either by a combinational circuit or by a look up table. The algorithms require both time and space linear in m. There are two approaches to compute $F(X_n, \ldots, X_1)$.

1. Top-down : We start using X_n, and descend through the variables.

2. Bottom-up : We start from the variable X_{n-m} and ascend through the variables.

Both the algorithms are implemented using pipelined hardware architecture. Each block of the hardware can be realized either by logic circuits or by lookup tables. We use logic circuits for one implementation and lookup tables for another. The algorithms by themselves are interesting and have been implemented using C programming language.

7.9.1 Top-down Algorithm

We introduce some notation to describe the algorithm. *It is important to remember that* $n = k + m + 1$ *throughout this discussion.* Let F be represented by $(h, S_1, \ldots, S_{m+1})$, where h is a function of $k = n - m - 1$ variables. Define $F_0 = h$ and F_i to be a function represented by $(h, S_1, \ldots, S_i)$. Then $F_{m+1} = F$, which we compute here. As $S_i = (\Psi_i, \tau_i)$, it is easy to see that the following holds.

$$
\begin{aligned}
F_i^{\tau_i} &= 1 \oplus F_i(X_{i+k}, X_{i+k-1}, \ldots, X_1) && \text{if } \tau_i = c, \\
F_i^{\tau_i} &= F_i(X_{i+k} \oplus 1, \ldots, X_1 \oplus 1) && \text{if } \tau_i = r \text{ and} \\
F_i^{\tau_i} &= 1 \oplus F_i(X_{i+k} \oplus 1, \ldots, X_1 \oplus 1) && \text{if } \tau_i = rc. \text{ Also,} \\
F_{i+1} &= (1 \oplus X_{i+k+1}) F_i(X_{i+k}, X_{i+k-1}, \ldots, X_1) \\
&\quad \oplus X_{i+k+1}(F_i(X_{i+k}, X_{i+k-1}, \ldots, X_1))^{\tau_{i+1}} && \text{if } \Psi_i = Q \text{ and} \\
F_{i+1} &= (1 \oplus X_{i+k}) F_i(X_{i+k+1}, X_{i+k-1}, \ldots, X_1) \\
&\quad \oplus X_{i+k}(F_i(X_{i+k+1}, X_{i+k-1}, \ldots, X_1))^{\tau_{i+1}} && \text{if } \Psi_i = R.
\end{aligned}
$$

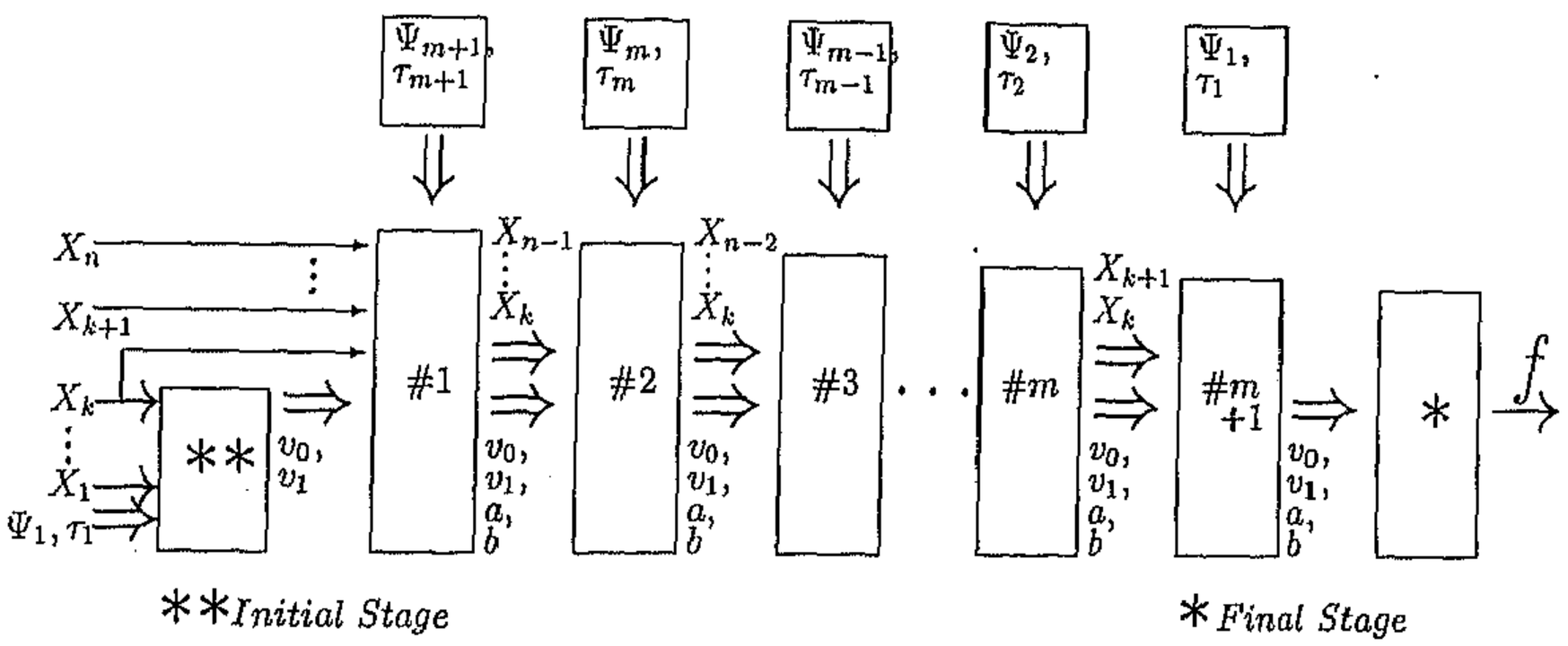


Figure 7.2: Top Down Architecture : Pipelined Implementation of $computeTD(.)$

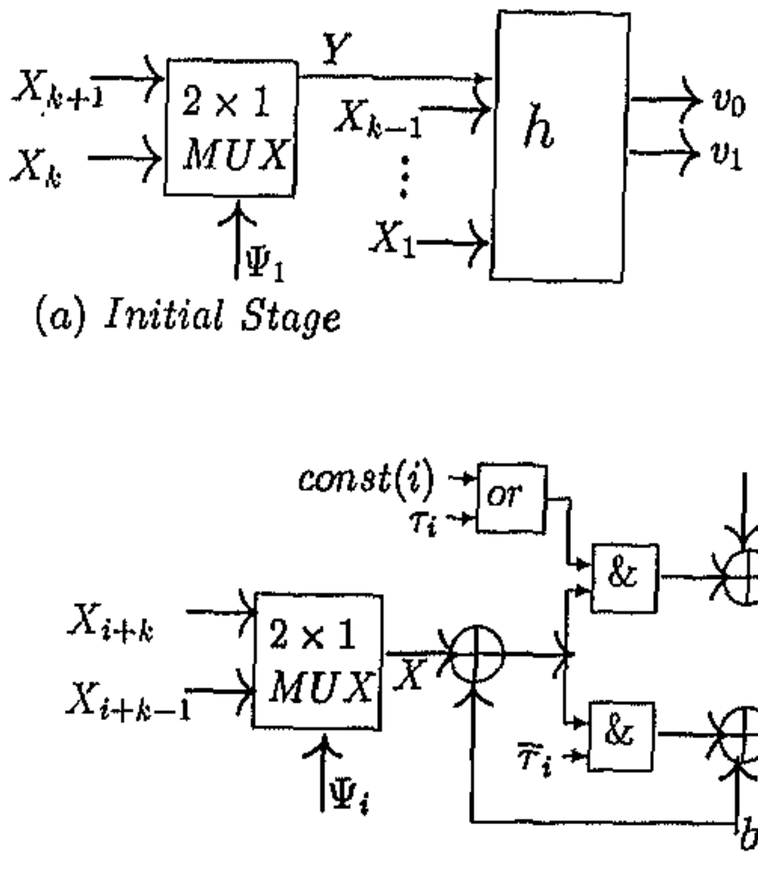
We provide an algorithm $computeTD(X_n, \ldots, X_1)$ based on the following two observations.

1. If $\Psi_i = R$, then in the next stage we have to interchange the role of X_{i+k} and X_{i+k-1}. See lines (1*) and (2*) in the algorithm $computeTD(.)$.

2. If ($\Psi_i = Q$ and $X_{i+k} = 1$) or ($\Psi_i = R$ and $X_{i+k-1} = 1$) then depending on τ_i, in the next stage either the output is complemented or the input is complemented or both output and input are complemented. This can be tracked by using two bit variables a and b, one for the output and one for the input.

We next present an algorithm for $computeTD(.)$.

$computeTD(X_n, \ldots, X_1)$ {
  if $(\Psi_1 = Q)$ then $Y = X_k$;
  if $(\Psi_1 = R)$ then $Y = X_{k+1}$;
  $v_0 = h(Y, X_{k-1}, \ldots, X_1)$; $v_1 = h(1 \oplus Y, 1 \oplus X_{k-1}, \ldots, 1 \oplus X_1)$;
  $a = 0$; $b = 0$;
  for $i = m + 1$ downto 1 do {
  (1*) if $(\Psi_i = Q)$ then $X = X_{i+k}$;
  (2*) if $(\Psi_i = R)$ then { $X = X_{i+k-1}$; $X_{i+k-1} = X_{i+k}$; }
  if $(b \oplus X = 1)$ then {
    if $(\tau_i = c)$ then $a = a \oplus 1$;
    if $(\tau_i = r)$ then $b = b \oplus 1$;



(a) *Initial Stage*

(c) *Final Stage*

(b) *Intermediate Stage*

Figure 7.3: Components of Top Down Architecture

```
        if (τ_i = rc) then {a = a ⊕ 1; b = b ⊕ 1; }
    }
    return a ⊕ v_b;
}
```

The following is clear from the previous discussion.

Theorem 7.9.1 *The algorithm computeTD*$(X_n, \ldots, X_1)$ *correctly computes* $F(X_n, \ldots, X_1)$ *in* $O(m)$ *time.*

Next we show how very low cost pipelined hardware can be developed where the output of F on successive tuples of n-bit input is available at each clock pulse after initial $(m+1)$ clocks, i.e. starting from $(m+2)$th clock.

7.9.2 Hardware Implementation of *computeTD*(.)

First we describe the encoding of Ψ, τ.

$\Psi_i = Q = 0$; $\tau_i = r = 0$ if i even; $\tau_i = rc = 0$ if i odd;
$\Psi_i = R = 1$; $\tau_i = c = 1$ if i even; $\tau_i = c = 1$ if i odd;

There are $(m+1)$ stages numbered #1 to #$(m+1)$ (see Figure 7.2). Stage #i stores the

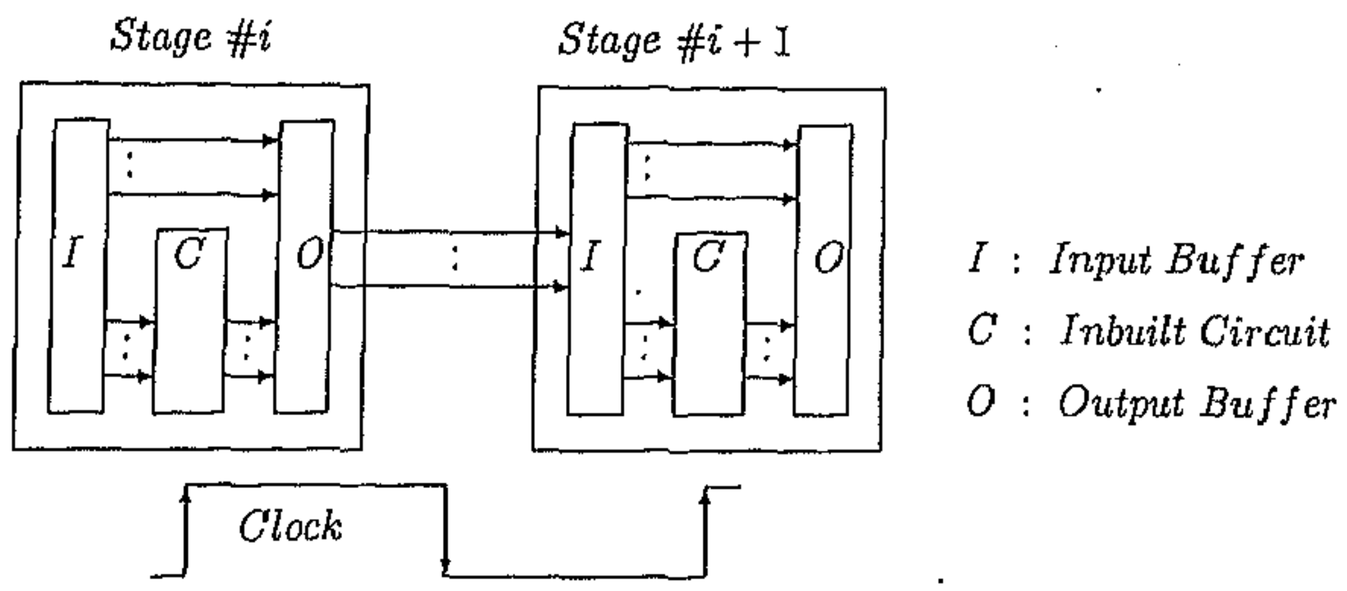


Figure 7.4: Input Output Latching for Intermediate Stages

current values of $X_k, \ldots, X_{n-i+1}$. The two bits v_0 and v_1 are present at each stage alongwith two other work bits a and b. If $\Psi_i = R$ the value of X_{i+k} and X_{i+k-1} should be properly interchanged for the next stage as in lines (1*) and (2*) of the algorithm $computeTD(.)$ and this is also implemented in the Figure 7.3c using MUX. The computation done in stage $\#i$ is to determine the values of a and b of stage $\#(i+1)$ and the circuit to do so is shown in Figure 7.3b as the intermediate circuit. Note that $const(i) = i \bmod 2$ has a fixed value for stage $\#i$. This is required to identify the arity of the stage $\#i$ to choose between r and rc for τ according as Construction 1. The initial stage (Figure 7.3a) determines the values v_0 and v_1 and the final stage (Figure 7.3c) performs the computation $a \oplus v_b$.

The whole circuit operates as follows. At each clock, stage $\#i$ forwards the the values of the variables to the next stage and updates the values of work bits a, b for the next stage. The values v_0 and v_1 are forwarded unchanged. It is important to understand the need for generation of v_0, v_1 at the first stage and carrying them through all the $m+1$ stages. We need these two bits only at the end for the final circuit (Figure 7.3c). However, the values of v_0, v_1 are generated from the variables X_1 to X_{k+1}. It is more efficient to carry two bits v_0, v_1 through the $m+1$ stages instead of carrying the $k+1$ bits $X_1, \ldots, X_{k+1}$. Since there are $m+1$ stages, the whole pipeline takes $m+1$ clocks to be completely filled up. Hence the first output appears at $(m+2)$th clock and consequently a bit of output appears at each clock.

Note that we use both the rising and falling edge of the clock. Each stage stores two buffers, one input and another output (see Figure 7.4). At the leading edge the values of the input buffer registers of stage $\#i$ are latched to the output buffer registers of the same stage. The signals $X_k, \ldots, X_{i+k-2}$ and v_0, v_1 go directly from input buffer to output buffer. The other three signals X_{i+k-1}, a, b are generated through the inbuilt combinational circuit

Value of X_{i+k}	$\Psi_i = Q, \tau_i = r$	$\Psi_i = Q, \tau_i = c$	$\Psi_i = Q, \tau_i = rc$
0	Fu = fu; Fur = fu; Fl = fur; Flr = fur;	Fu = fu; Fur = 1 - fur; Fl = 1 - fu; Flr = fur;	Fu = fu; Fur = 1 - flu; Fl = 1 - fur; Flr = fur;
1	Fu = fur; Fur = fur; Fl = fu; Flr = fu;	Fu = 1 - fu; Fur = fur; Fl = fu; Flr = 1 - fur;	Fu = 1 - fur; Fur = fur; Fl = fu; Flr = 1 - fu;

Value of X_{i+k} and X_{i+k-1}	$\Psi_i = R, \tau_i = r$	$\Psi_i = R, \tau_i = c$	$\Psi_i = R, \tau_i = rc$
0, 0	Fu = fu; Fur = fu; Fl = fl; Flr = fl;	Fu = fu; Fur = 1 - fur; Fl = fl; Flr = 1 - flr;	Fu = fu; Fur = 1 - fu; Fl = fl; Flr = 1 - fl;
0, 1	Fu = flr; Fur = flr; Fl = fur; Flr = fur;	Fu = 1 - fl; Fur = flr; Fl = 1 - fu; Flr = fur;	Fu = 1 - flr; Fur = flr; Fl = 1 - fur; Flr = fur;
1, 0	Fu = fl; Fur = fl; Fl = fu; Flr = fu;	Fu = fl; Fur = 1 - flr; Fl = fu; Flr = 1 - fur;	Fu = fl; Fur = 1 - fl; Fl = fu; Flr = 1 - fu;
1, 1	Fu = fur; Fur = fur; Fl = flr; Flr = flr;	Fu = 1 - fu; Fur = fur; Fl = 1 - fl; Flr = flr;	Fu = 1 - fur; Fur = fur; Fl = 1 - flr; Flr = flr;

Table 7.2: Table for calculating bit variables

(Figure 7.3b) from X_{i+k}, X_{i+k-1}, a, b and Ψ_i, τ_i. That is, the stage C in Figure 7.4 contains the circuit of Figure 7.3c. At the falling edge of the clock, the output buffer registers of stage $\#i$ are latched to the input buffer registers of the stage $\#(i+1)$. The inbuilt combinational circuit being small enough, it is justified to consider that the delay of the circuit is much less than the clock width and hence there is no problem in using both the leading and falling edge of the clock in the hardware. The inbuilt combinational circuit blocks in this architecture can also be implemented using lookup table. We use that approach for the bottom up implementation.

7.9.3 Bottom-up Algorithm

We consider the bottom up algorithm in this section. Let us recapitulate the Definition 7.2.2.
Let $f, g \in \Omega_{n-1}$, and $F \in \Omega_n$. If $F = Q(f, g)$ then
$F(X_n, X_{n-1}, \ldots, X_1) = (1 \oplus X_n)\ f(X_{n-1}, \ldots, X_1) \oplus X_n\ g(X_{n-1}, \ldots, X_1)$.
If $F = R(f, g)$ then $F(X_n, X_{n-1}, \ldots, X_1) =$
$(1 \oplus X_n)(1 \oplus X_{n-1})\ f(X_{n-1} = 0, X_{n-2}, \ldots, X_1) \oplus (1 \oplus X_n)X_{n-1}\ g(X_{n-1} = 0, X_{n-2}, \ldots, X_1) \oplus$
$X_n(1 \oplus X_{n-1})\ f(X_{n-1} = 1, X_{n-2}, \ldots, X_1) \oplus X_n X_{n-1}\ g(X_{n-1} = 1, X_{n-2}, \ldots, X_1)$.
To carry the information of the lower stage to upper stage we use four variables as follows.
For $h \in \Omega_k$, let $fu = h(X_k, X_{k-1}, \ldots, X_2, X_1)$, $fl = h(1 \oplus X_k, X_{k-1}, \ldots, X_2, X_1)$,
$fur = h(1 \oplus X_k, 1 \oplus X_{k-1}, \ldots, 1 \oplus X_2, 1 \oplus X_1)$, and $flr = h(X_k, 1 \oplus X_{k-1}, \ldots, 1 \oplus X_2, 1 \oplus X_1)$.

We start with four different bit variables fu, fl, fur, flr for given values of $X_1, X_2, \ldots, X_{k-1}, X_k$. Then for each step i $(1 \leq i \leq m+1)$ we update the values of these four variables. It is important to note that by manipulating only these four variables alongwith the values of at most two input variables at each step, the final output of the function is generated. We use four temporary variables Fu, Fl, Fur, Flr and the Table 7.2 in the algorithm.

computeBU$(X_n, \ldots, X_1)$ {

$fu = h(X_k, X_{k-1}, \ldots, X_2, X_1)$;
$fl = h(1 \oplus X_k, X_{k-1}, \ldots, X_2, X_1)$;
$fur = h(1 \oplus X_k, 1 \oplus X_{k-1}, \ldots, 1 \oplus X_2, 1 \oplus X_1)$;
$flr = h(X_k, 1 \oplus X_{k-1}, \ldots, 1 \oplus X_2, 1 \oplus X_1)$;

for $(i = 1, i \leq m+1, i++)$ {
Assign Fu, Fl, Fur, Flr using Table 7.2 from $X_{i+k}, X_{i+k-1}, \Psi_i, \tau_i, fu, fl, fur, flr$;
$fu = Fu$; $fl = Fl$; $fur = Fur$; $flr = Flr$;
}
return fu;
}

From the algorithm *computeBU*(.) it is clear that the initialization is of one step and then there are $m+1$ more steps to get the final output. Also the correctness of the above algorithm can be checked. Thus, we get the following theorem.

Theorem 7.9.2 *The algorithm computeBU*$(X_n, \ldots, X_1)$ *correctly computes* $F(X_n, \ldots, X_1)$ *in* $O(m)$ *time.*

7.9.4 Hardware Implementation of *computeBU*(.)

We map the algorithm to another pipelined architecture (see Figure 7.5) so that output can be available at each clock after initial $m+1$ clocks.

At the first clock, the function h, depending on the k inputs $\{X_1, X_2, \ldots : X_k\}$, will send a four bit output fu, fl, fur, flr to the first stage of the pipeline. Also the input variables $\{X_k, X_{k+1}, \ldots X_n\}$ will be sent to the first stage directly.

The bits fu, fl, fur, flr are the work bits for each stage. At the second clock, another 4 bit output from h alongwith the new inputs will again go to the first stage of the pipeline. The first stage of the pipeline, depending on the values of (Ψ_1, τ_1), inputs X_k, X_{k+1} and fu, fl, fur, flr will send new set of fu, fl, fur, flr to the second stage of the pipeline alongwith the input variables $\{X_{k+1}, X_{k+2}, \ldots X_n\}$. Note that h is an n input 1 output Boolean

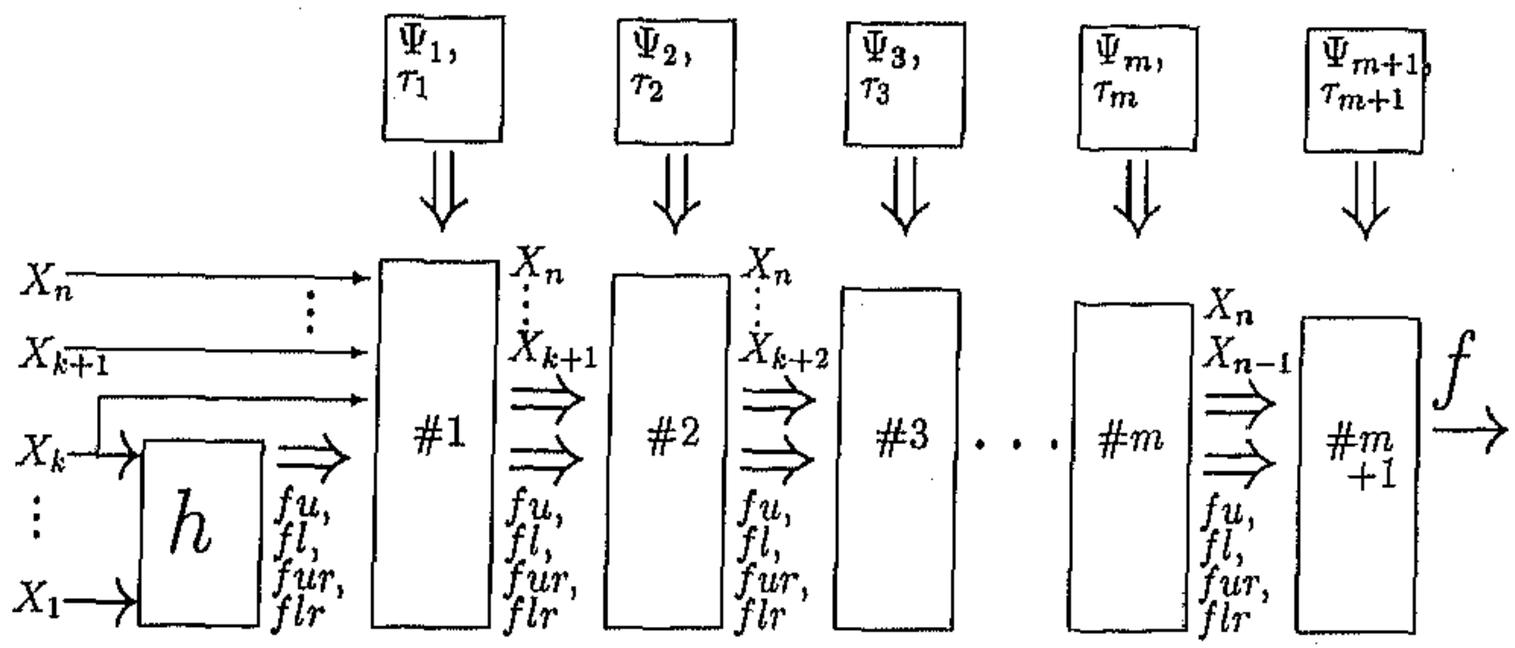


Figure 7.5: Bottom Up Architecture : Pipelined Implementation of *computeBU*(.)

function. However, at each clock we need four output bits fu, fl, fur, flr from the hardware implementation of h, i.e. in hardware h will produce a four bit output according to the initialization step of the algorithm *computeBU*().

In this manner, at the $(m+1)$th clock, the values fu, fl, fur, flr will reach to the $(m+1)$th stage of the pipeline alongwith the input variables X_{n-1}, X_n and we will get the first output at the $(m+2)$th clock. From the next clock onwards, the output of f will be available at each clock.

We use the same kind of input and output buffering at the rising and falling edge of the clock which we have used in the hardware for *computeTD*(.). The blocks containing (Ψ, τ) can be considered as the control unit to the hardware. Given fixed h, depending on the values of the control unit, the exact function will be realized. Next we estimate the number of register bits required for this circuit. The control parameters Ψ, τ are represented by two bits as in the circuit description of top down approach. Thus, for $(m+1)$ stages $2(m+1)$ bits are necessary. At each stage fu, fl, fur, flr will take 4 bits. Also these are to be stored in both the input and output buffers of each stage. So we need $2 \times 4 \times (m+1) = 8(m+1)$ register bits.

Let us now consider the number of bits required in the input buffer at each stage to store the input variables. In the first stage the input variables stored is from X_k to X_n, i.e. total $n-k+1 = m+2$ bits are required, whereas in the last stage it is only 2 bits to store X_{n-1} and X_n. So total space required to store the input variables $= (m+2)+(m+1)+\ldots+3+2 = \frac{(m+2)(m+3)}{2} - 1 = \frac{m^2+5m+4}{2}$. Similarly for the output buffer the required number of bits is

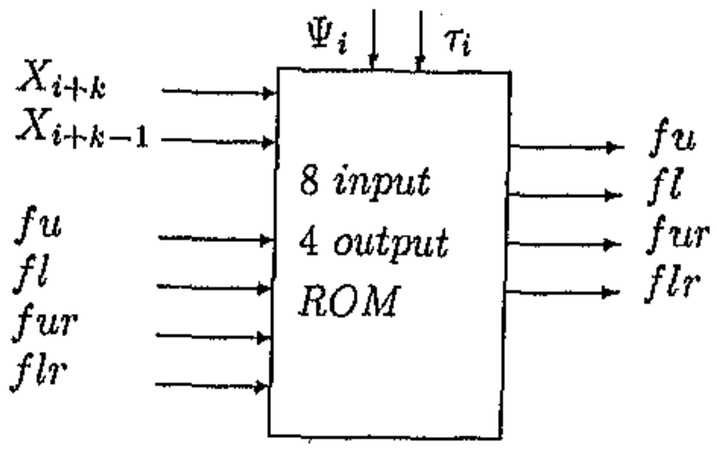


Figure 7.6: Look up Table for Intermediate Stage #i

input	output (even i)	output (odd i)
00000000	0000	0110
00000100	0110	0110

input	output (even i)	output (odd i)
00001000	0000	0101
00001100	0101	0101

Table 7.3: Input Output relations of lookup tables

$(m+1)+m+\ldots+2+1=\frac{(m+1)(m+2)}{2}=\frac{m^2+3m+2}{2}$. Thus the total register bit requirement for input and output buffer $=\frac{m^2+5m+4}{2}+\frac{m^2+3m+2}{2}=m^2+4m+3$. Thus overall we need $2(m+1)+8(m+1)+(m^2+4m+3)=m^2+14m+13$ register bits.

Next we consider the complexity of the circuit in each stage of the pipeline. The circuit consists of generating four output bits fu, fl, fur, flr from $X_{i+k-1}, X_{i+k}, \Psi_i, \tau_i$ and fu, fl, fur, flr of earlier stage, i.e. from total of 8 bits. This can be easily implemented by an 8-bit input and 4-bit output lookup table which takes $4 \times 2^8 = 1024$ bits. The lookup table, Table 7.2, can be easily implemented using a ROM as in Figure 7.6. Each ROM has 256 addresses and each address stores 4 bits of data. Note that for $\Psi = Q$, the input X_{i+k-1} is not required. However, it is simpler to consider same number of input lines. Note that the lookup tables for even numbered stages and odd numbered stages are different. This is due to the fact that $\tau_i = 0$ represents r if i is even and $\tau_i = 0$ represents rc if i is odd. Let, input to the lookup table be 8 bits $< fu, fur, fl, flr, \Psi_i, \tau_i, X_{i+k}, X_{i+k-1} >$ and output be 4 bits $< fu, fur, fl, flr >$ for the next stage. Table 7.3 provides few examples of input output relations for the two types of lookup tables, for i even and i odd. For $m+1$ stages the overall requirement is of $1024 \times (m+1)$ bits. Design of this kind of circuit using basic logic gates is also simple. Direct implementation of an n variable Boolean function using lookup table takes 2^n bits. The method we consider here uses the following parts.

1. Implementing h of k variables takes 4×2^k bits using look up table.
2. Each of the $m+1$ stages takes 1024 bits if implemented using look up table.
3. $m^2 + 14m + 13$ register bits are required to store input variables, control bits and the

work bits considering all stages.
Thus the total requirement is $4 \times 2^k + 1024 \times (m+1) + m^2 + 14m + 13$ bits instead of 2^n bits for implementing lookup table, where $n = k + m + 1$. This provides an enormous saving of hardware space and makes it possible to implement cryptographically significant Boolean functions on large number of input variables. Associated control units and the store and forward kind of architecture is also simple to implement.

The method proposed here can be used effectively to construct functions with large number of variables. As an example, if we consider $h \in \Omega_{20}$ and construct f represented by a sequence $(h, S_1, \ldots, S_{30})$ satisfying Theorem 7.6.2, then $f \in \Theta_{50}(29)$ with nonlinearity $2^{49} - 2^{39} - 2^{30}$ and degree 20 can be achieved. *Currently, there are no known methods which can construct such an optimized function with better or even equal nonlinearity. Moreover, our functions can be implemented with nominal hardware.* Direct implementation of f using truth table will take 2^{50} bits, which is not feasible to store. However, using the representation of f as $(h, S_1, \ldots, S_{30})$, it is possible to implement f with much less hardware. The implementation of h needs 4×2^{20} bits, i.e. 4 Megabits by implementing h with truth table. Since $m = 29$, we need $m^2 + 14m + 13 = 1260$ more register bits and associated lookup table for 30 stages which is equal to $30 \times 1024 = 30720$ bits.

The reconfigurability of the above architecture can be achieved as follows. The function can be changed by changing the values of the register bits (Ψ, τ) in the control units. Suppose $h \in \Omega_k$ is fixed. In each stage there are 4 options for each (Ψ, τ) pair. Thus, for $m+1$ stages there are total 4^{m+1} options. Hence for a fixed h, 4^{m+1} possible functions can be realized. Implementing each function requires programming the total $2(m+1)$ control bits in $m+1$ stages. This allows using a wide range of functions with high nonlinearity which optimize Siegenthaler's inequality.

In LFSR based stream cipher system, the private key is the initial conditions of the LFSRs. In the architecture we develop here, the key may also contain the bit pattern in $m+1$ pairs of (Ψ, τ), i.e., additional $2(m+1)$ bits alongwith the initial conditions of the LFSRs. The inclusion of the control bits in the key makes the encryption scheme more secured. Even if the attacker has knowledge about the combining function at certain point of time, the function itself can be reconfigured easily for future communication, which makes the information of the attacker useless. Also it is important to note that the design proposed in this section is scalable which is helpful for VLSI design.

Here we have presented a design strategy for highly nonlinear balanced functions optimizing the algebraic degree and the order of correlation immunity according to Siegenthaler's bound. The recursive design methodology is suitable to implement such a Boolean function using pipelined architecture and the implementation is possible with nominal hardware even for large number of input variables. Such a balanced function with large number of input variables provides high nonlinearity alongwith high algebraic degree and high order

of correlation immunity. Use of this kind of function in stream cipher system makes the cryptographic scheme immune from all the currently known attacks.

Chapter 8

Concluding Remarks

Here we first summarize the contribution of the thesis and then list related open problems in this direction of research.

8.1 Summary

In this dissertation we have studied enumeration and construction problems of cryptographically significant Boolean functions. The following properties of Boolean functions are considered here.

- Balancedness
- Nonlinearity
- Algebraic Degree
- Correlation Immunity
- Symmetry
- Strict Avalanche Criteria
- Propagation Characteristics

The study contains combinatorial aspects of Boolean functions with these properties. These functions have important applications in design of private key cryptosystems. We also discuss the implementation aspect of some of these functions in hardware. Moreover, we study

theoretical aspects of such constructions which include enumeration and identification of new sets.

Chapter 1 presents an introduction. In Chapter 2, we provide a brief outline of existing research and show how our work fits in that framework.

In Chapter 3 we concentrate on symmetric Boolean functions. We discuss results related to the nonlinearity of symmetric functions. A closed form expression for the Walsh transform of an arbitrary symmetric Boolean function is presented in Theorem 3.2.1. We completely characterize the symmetric functions with maximum nonlinearity and show that the maximum nonlinearity of n-variable symmetric function can be $2^{n-1} - 2^{\lfloor \frac{n-1}{2} \rfloor}$. The important results for even n are Theorem 3.3.2, Theorem 3.3.4 and for odd n are Theorem 3.4.1, Theorem 3.4.2, Theorem 3.4.3. Moreover, new classes of symmetric balanced and symmetric correlation immune functions are presented in Section 3.5, Section 3.6.

It is well known that the Boolean functions to be used in cryptographic applications should have the balancedness property. In Chapter 4 we initially provide a randomized heuristic to construct balanced Boolean functions on n variables ($n \geq 15$ and odd) with nonlinearity strictly greater than $2^{n-1} - 2^{\frac{n-1}{2}}$. For such functions the algebraic degree is also maintained at its maximum, $n-1$ (see Theorem 4.2.1). For odd $n \leq 13$, in Theorem 4.2.3, we construct balanced functions with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ and algebraic degree $n-1$. Moreover, we design optimized 1-resilient functions with nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ (Theorem 4.3.3) for odd n. In Section 4.4 of this chapter we consider propagation characteristics and strict avalanche criteria. Our constructions provide balanced functions with these properties and maintain very high nonlinearity which could not be achieved earlier.

The set of correlation immune Boolean functions can be partitioned into several disjoint sets with respect to the Hamming weights of their output column. In Chapter 5 it is shown that the number of n variable correlation immune functions of Hamming weight $2a + 2$ is strictly greater than the number of such functions of weight $2a$ for $2a < 2^{n-1}$. The proof of this seemingly intuitive result (Theorem 5.2.3) turns out to be quite involved and we need a series of results to prove it. The technique also relates the enumeration problem of correlation immune functions to the enumeration problem of balanced correlation immune functions (Theorem 5.3.2) and provides a closed form expression on the number of correlation immune functions (Theorem 5.4.2).

However, the closed form expression can not be evaluated efficiently and it seems that counting the exact number of correlation immune functions using constructive methods is a nontrivial problem. That is the reason we concentrate on some small subsets of correlation immune functions. We estimate the cardinality of these small subsets in Theorem 6.3.3. Results using sharper necessary conditions are presented in Theorem 6.4.3. Further, functions are obtained (Section 6.5) for the set of correlation immune functions which satisfy one or more of a few other conditions e.g. balancedness, nondegeneracy and nonaffinity. Our tech-

niques clearly highlight the difficulty of exactly enumerating the set of correlation immune Boolean functions using constructive methods.

In Chapter 7 we provide a new construction method using a small set of recursive operations (Definition 7.2.2) for a large class of highly nonlinear, resilient Boolean functions optimizing Siegenthaler's inequality (Section 7.6). Comparisons to previous publications show that better nonlinearity can be obtained by our method. Our technique can be used to construct functions on large number of input variables with simple hardware implementation (Section 7.9). We provide a special representation for such functions so that they can be implemented with low cost pipelined architecture. Moreover, the architecture is programmable and can be dynamically reconfigured to compute different functions of the class.

8.2 Open Problems

In Chapter 3 we discussed symmetric Boolean functions. The following open problems are interesting in this direction.

1. A general Boolean function needs 2^n bits to be represented in the truth table format. The time complexity to calculate Walsh transform of such a function is $O(n2^n)$, i.e. $O(2^n \log_2 2^n)$ (fast Walsh transform algorithm). A symmetric Boolean function can be represented by $(n+1)$ bits and we provide an $O(n^3)$ algorithm for the Walsh transform. It would be interesting to find out whether it is possible to design a faster algorithm with time complexity $O(n \log_2 n)$ for the symmetric functions.

2. We have presented new constructions of symmetric balanced and symmetric correlation immune functions. Further constructions of such functions will be interesting and they will also require new identities over binomial coefficients. Moreover, it would be very important to find out how all the functions of these two classes can be characterized.

In Chapter 4 we provide constructions of balanced functions with important cryptographic properties.

1. We provide a randomized heuristic to get balanced Boolean functions on n variables ($n \geq 15$ and odd) with nonlinearity strictly greater than $2^{n-1} - 2^{\frac{n-1}{2}}$. However, the question of mathematically constructing such functions remains open.

2. Further construction methods of balanced PC(l) of order k functions and balanced SAC(k) functions with better nonlinearity than we have provided would be an interesting area of research.

In Chapter 5 and Chapter 6 we have worked on enumeration of correlation immune functions. The exact enumeration problem of correlation immune functions is still unsolved. In Chapter 5 calculation of $C_n(2a)$ and $C_{n,x}(2a)$ are open. It will also be interesting to get a good estimate of the number of correlation immune Boolean functions using constructive methods which will be competitive to the result of Denisov [28].

We have discussed design and implementation issues of m-resilient functions in Chapter 7 and provide better nonlinearity than all the previous efforts. However, we are presently working in this area and it is important to point out the current results [99, 98, 101, 115, 87] before posing the open problems. The current target is finding out m-resilient Boolean functions on n variables with algebraic degree $n-m-1$ and more importantly with provably maximum nonlinearity. In this direction we first mention important results on Walsh spectra of resilient and correlation immune functions [101].

(i) Let f be an n-variable, m-resilient (with $n \geq 3$ and $m \leq n-3$) function and $l \in L(n)$. Then $d(f,l)$ (respectively $wd(f,l)$) is congruent to 0 mod 2^{m+1} (respectively 0 mod 2^{m+2}).

(ii) Let f be an n-variable, m-th order correlation immune (with $n \geq 3$ and $m \leq n-2$) function and $l \in L(n)$. Then $d(f,l)$ (respectively $wd(f,l)$) is congruent to 0 mod 2^m (respectively 0 mod 2^{m+1}).

Next we provide the results related to upper bound of nonlinearity of m-resilient and m-th order correlation immune Boolean functions [101]. Note that by $nlmax(n)$ we mean the maximum possible nonlinearity of any n variable Boolean function when n is odd.

(I) Let $nlR(n,m)$ be the maximum possible nonlinearity of n-variable m-resilient functions.
(i) If n is even and $m+1 > \frac{n}{2}-1$, then $nlR(n,m) \leq 2^{n-1} - 2^{m+1}$.
(ii) If n is even and $m+1 \leq \frac{n}{2}-1$, then $nlR(n,m) \leq 2^{n-1} - 2^{\frac{n}{2}-1} - 2^{m+1}$.
(iii) If n is odd and $2^{m+1} > 2^{n-1} - nlmax(n)$, then $nlR(n,m) \leq 2^{n-1} - 2^{m+1}$.
(iv) If n is odd and $2^{m+1} \leq 2^{n-1} - nlmax(n)$, then $nlR(n,m)$ is the highest multiple of 2^{m+1} which is less than or equal to $2^{n-1} - nlmax(n)$.
Further in cases (i) and (iii), the spectra of any function achieving the stated bound must be three valued, i.e. the values of the Walsh distances must be $0, \pm 2^{m+2}$.

(II) Let $nlc(n,m)$ denote the highest possible nonlinearity for an n-variable function which is CI of order m. Then we have the following.
(i) If n is even and $m > \frac{n}{2}-1$, then $nlc(n,m) \leq 2^{n-1} - 2^m$.
(ii) If n is even and $m \leq \frac{n}{2}-1$, then $nlc(n,m) \leq 2^{n-1} - 2^{\frac{n}{2}-1} - 2^m$.
(iii) If n is odd and $2^m > 2^{n-1} - nlmax(n)$, then $nlc(n,m) \leq 2^{n-1} - 2^m$.
(iv) If n is odd and $2^m \leq 2^{n-1} - nlmax(n)$, then $nlc(n,m)$ is the highest multiple of 2^m which is less than or equal to $2^{n-1} - nlmax(n)$.
Further in cases (i) and (iii), the spectra of any function achieving the stated bound must be three valued, i.e. the values of the Walsh distances must be $0, \pm 2^{m+1}$.

	1	2	3	4	5	6	7	8
5	12	8	0					
6	24	24	16	0				
7	56	56	48	32	0			
8	112(116)	112	112	96	64	0		
9	240(244)	232(240)	224(240)	224	192	128	0	
10	484(492)	480(488)	480	480	448	384	256	0

Table 8.1: Nonlinearity of resilient Boolean functions on small number of variables

It is still not clear whether in all the cases the upper bounds can be achieved or not. However, it is possible to find out large subsets of resilient functions for which these upper bounds are achievable. Let us first tabulate the current scenario for resilient Boolean functions on small number of variables. Here we tabulate the maximum possible nonlinearity for degree optimized resilient functions, which can be constructed. The columns represent the order of resiliency and the rows represent the number of variables. In cases where the upper bound has not yet been achieved, we write the upper bound value in the parenthesis. *Constructing such functions or proving their nonexistence will be interesting research problems.* Generalized construction of such functions is possible by using different kinds of linear function concatenation techniques. Camion et al [12] has initially provided a construction method of resilient functions. The same subset has been examined by Seberry et al [105] and new directions has been provided to consider the algebraic degree and nonlinearity of such constructions. In Section 7.7 of Chapter 7 (see also [67]), we interpret the same construction as concatenation of linear functions. This linear function concatenation technique has consequently been sophisticated and we are getting better results in terms of nonlinearity of resilient functions [99, 98, 101]. However, these new techniques are not sufficient to characterize the set of optimal resilient functions (i.e. which optimize Siegenthaler's inequality and provide provably maximum nonlinearity) in all the cases.

Let (n, m, d, x) denotes an n-variable, m-resilient Boolean function with algebraic degree d and nonlinearity x. Let us consider the problem of constructing $(3+2i, i, 2+i, 2^{2+2i}-2^{1+i})$ functions. For $i = 1$, the $(5, 1, 3, 12)$ function is possible to construct. *However, for $i \geq 2$, the construction problem remains open. As example, for $i = 2$, the $(7, 2, 4, 56)$ function and for $i = 3$, the $(9, 3, 5, 240)$ function are two interesting objects of future study.* In [101], we show that it is possible to construct $(3 + 2i + j, i + j, 2 + i, 2^{2+2i+j} - 2^{1+i+j})$ functions for all $j \geq t_0$ where $2^{1+i} = 3 + i + t_0$. These functions possess the provably maximum possible nonlinearity. *However, the construction of $(3 + 2i + j, i + j, 2 + i, 2^{2+2i+j} - 2^{1+i+j})$ functions with $j < t_0$ and $i \geq 3$ is a challenging research problem.* Note that these are the functions

with higher order of resiliency, i.e., when $m > \lfloor \frac{n}{2} \rfloor - 2$.

The problem seems to be more complicated for lower order of resiliency. In Chapter 4 we have provided methods to construct optimized 1-resilient functions with nonlinearity $2^{n-1} - 2^{\lfloor \frac{n}{2} \rfloor}$. Recently we have proved [100] that given any order of resiliency m, it is possible to find an n such that there exists an n-variable, m-resilient Boolean function with nonlinearity strictly greater than $2^{n-1} - 2^{\lfloor \frac{n}{2} \rfloor}$. However, it is not clear what is the maximum possible achievable nonlinearity for such functions. In particular, the functions on small number of input variables such as $(8, 1, 6, 116)$, $(9, 1, 7, 244)$, $(10, 1, 8, 492)$ seems hard to construct. Also it is not proved whether these functions do not exist.

Bibliography

[1] C. M. Adams and S. E. Tavares. Generating and counting binary bent sequences. *IEEE Transactions on Information Theory*, IT-36(8):1170–1173, September 1990.

[2] K. Beauchamp. *Applications of Walsh and Related Functions.* Academic Press, 1984.

[3] E. R. Berlekamp. *Algebraic Coding Theory.* McGraw Hill, New York, 1968.

[4] 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, January 1972.

[5] J. Bierbrauer, K. Gopalakrishnan, and D. R. Stinson. Bounds on resilient functions and orthogonal arrays. In *Advances in Cryptology - CRYPTO'94*, number 839 in Lecture Notes in Computer Science, pages 247–256. Springer Verlag, 1994.

[6] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. In *Advances in Cryptology - CRYPTO'90*, Lecture Notes in Computer Science, pages 2–21. Springer-Verlag, 1991.

[7] L. Blum, M. Blum, and M. Shub. A simple unpredictable random number generator. *SIAM Journal on Computing*, 15:364–383, 1986.

[8] E. F. Brickell, J. H. Moore, and M. R. Purtill. Structures in the S-boxes of the DES. In *Advances in Cryptology - CRYPTO'86*, Lecture Notes in Computer Science, pages 3–8. Springer-Verlag, 1987.

[9] R. A. Brualdi, N. Cai, and V. S. Pless. Orphan structures of first order Reed-Muller codes. *Discrete Mathematics*, (102):239–247, 1992.

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

[11] J. O. Brüer. On pseudorandom sequences as crypto generators. In *International Zurich Seminar on Digital Communications*, pages 157–161. IEEE, New York, 1984.

[12] P. Camion, C. Carlet, P. Charpin, and N. Sendrier. On correlation immune functions. In *Advances in Cryptology - CRYPTO'91*, pages 86–100. Springer-Verlag, 1992.

[13] A. Canteaut and E. Filiol. Ciphertext only reconstruction of stream ciphers based on combination generators. In *Fast Software Encryption'2000*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

[14] C. Carlet. Partially bent functions. *Designs, Codes and Cryptography*, 3:135–145, 1993.

[15] C. Carlet. Two new classes of bent functions. In *Advances in Cryptology - EUROCRYPT'93*, number 765 in Lecture Notes in Computer Science, pages 77–101. Springer-Verlag, 1994.

[16] C. Carlet. Recent results on binary bent functions. In *International Conference on Combinatorics, Information Theory and Statistics*, 1997.

[17] C. Carlet and P. Guillot. A characterization of bent functions. *Journal of Combinatorial Theory, Series A*, 76(2):328–335, September 1996.

[18] S. Chee, S. Lee, and K. Kim. Semi-bent functions. In *Advances in Cryptology, Asiacrypt'94*, number 917 in Lecture Notes in Computer Science, pages 107–118. Springer-Verlag, 1995.

[19] S. Chee, S. Lee, D. Lee, and S. H. Sung. On the correlation immune functions and their nonlinearity. In *Advances in Cryptology, Asiacrypt 96*, number 1163 in Lecture Notes in Computer Science, pages 232–243. Springer-Verlag, 1996.

[20] V. Chepyzhov and B. Smeets. On a fast correlation attack on certain stream ciphers. In *Advances in Cryptology - EUROCRYPT'91*, volume 547, pages 176–185. Springer-Verlag, 1991.

[21] B. Chor, O. Goldreich, J. Hastad, J. Friedman, S. Rudich, and R. Smolensky. The bit extraction problem or t-resilient functions. In *26th IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985.

[22] A. Clark, J. Golic, and E. Dawson. A comparison of fast correlation attacks. In *Fast Software Encryption, FSE'96*, volume 1039, pages 145–158. Springer-Verlag, 1996.

[23] G. D. Cohen, M. G. Karpovsky, H. F. Mattson, and J. R. Schatz. Covering radius - survey and recent results. *IEEE Transactions on Information Theory*, IT-31(3):328–343, 1985.

[24] T. W. Cusick. Boolean functions satisfying higher order strict avalanche criterion. In *Advances in Cryptology - EUROCRYPT'93*, number 765 in Lecture Notes in Computer Science, pages 102–117. Springer-Verlag, 1994.

[25] L. Dadda and V. Piuri. Pipelined adders. *IEEE Transactions on Computers*, 45(3):348–356, March 1996.

[26] E. Dawson and C. K. Wu. Construction of correlation immune Boolean functions. In *Information and Communications Security*, Lecture Notes in Computer Science, pages 170–180. Springer-Verlag, 1997.

[27] J. D. Denev and V. D. Tonchev. On the number of equivalence classes of Boolean functions under a transformation group. *IEEE Transactions on Information Theory*, IT-26(5):625–626, SEPTEMBER 1980.

[28] O. V. Denisov. An asymptotic formula for the number of correlation-immune of order k Boolean functions. *Discrete Mathematics and Applications*, 2(4):407–426, 1992.

[29] T. C. Denk and K. K. Parhi. Synthesis of folded pipelined architectures for multirate DSP algorithms. *IEEE Transactions on VLSI Systems*, 6(4):595–607, December 1998.

[30] W. Diffe and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(5):644–654, 1976.

[31] C. Ding, G. Xiao, and W. Shan. *The Stability Theory of Stream Ciphers*. Number 561 in Lecture Notes in Computer Science. Springer-Verlag, 1991.

[32] H. Dobbertin. Construction of bent functions and balanced Boolean functions with high nonlinearity. In *Fast Software Encryption*, number 1008 in Lecture Notes in Computer Science, pages 61–74. Springer-Verlag, 1994.

[33] E. Filiol and C. Fontaine. Highly nonlinear balanced Boolean functions with a good correlation-immunity. In *Advances in Cryptology - EUROCRYPT'98*. Springer-Verlag, 1998.

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

[35] R. Forre. The strict avalanche criterion : Spectral properties of Boolean functions and an extended definition. In *Advances in Cryptology - CRYPTO'88*, Lecture Notes in Computer Science, pages 450–468. Springer-Verlag, 1990.

[36] J. Friedman. On the bit extraction problem. In *33rd IEEE Symposium on Foundations of Computer Science*, pages 314–319, 1982.

[37] J. Dj. Golic, M. Salmasizadeh, L. Simpson, and E. Dawson. Fast correlation attacks on nonlinear filter generators. *Information Processing Letters*, 64(1):37–42, 1997.

[38] S. W. Golomb. *Shift Register Sequences.* San Fransisco, CA, Holden-Day, 1967.

[39] K. Gopalakrishnan. *A study of Correlation-immune, resilient and related cryptographic functions.* PhD thesis, University of Nebraska, 1994.

[40] 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.

[41] K. Gopalakrisnan and D. R. Stinson. Three characterizations of non-binary correlation-immune and resilient functions. *Designs, Codes and Cryptography*, 5:241–251, 1995.

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

[43] R. W. Hamming. *Coding And Information Theory.* Prentice Hall Inc., 1980.

[44] T. Honda, T. Satoh, T. Iwata, and K. Kurosawa. Balanced Boolean functions satisfying PC(2) and very large degree. In *SAC'97*, pages 64–72, January 1997.

[45] X. Hou. Covering radius of the Reed-Muller code $R(1,7)$ - a simpler proof. *Journal of Combinatorial Theory, Series A*, 74(3):337–341, 1996.

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

[47] K. Hwang. *Advanced Computer Architecture.* McGraw-Hill, 1993.

[48] T. Jakobsen and L. R. Knudsen. The interpolation attack on block ciphers. In *SAC'97*, pages 28–40, January 1997.

[49] T. Johansson. Reduced complexity correlation attacks on twoclock-controlled generators. In *Advances in Cryptology - ASIACRYPT'98*, Lecture Notes in Computer Science. Springer-Verlag, 1998.

[50] T. Johansson. A simple algorithm for fast correlation attacks on stream ciphers. In *Fast Software Encryption'2000*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

[51] T. Johansson and F. Jonsson. Fast correlation attacks based on turbo code techniques. In *Advances in Cryptology - CRYPTO'99*, number 1666 in Lecture Notes in Computer Science, pages 181–197. Springer-Verlag, August 1999.

[52] T. Johansson and F. Jonsson. Improved fast correlation attacks on stream ciphers via convolutional codes. In *Advances in Cryptology - EUROCRYPT'99*, number 1592 in Lecture Notes in Computer Science, pages 347–362. Springer-Verlag, May 1999.

[53] F. Klass, M. Flynn, and A. J. Van De Goor. Fast multiplication in VLSI using wave pipelining techniques. *Journal of VLSI Signal Processing*, 7:233–248, 1994.

[54] D. E. Knuth. *The Art of Computer Programming*, volume 2. Addison Wesley, 1969.

[55] I. Krasikov. On integral zeros of Krawtchouk polynomials. *Journal of Combinatorial Theory, Series A*, 74:71–99, 1996.

[56] S. Krishnakumar, P. Suresh, S. Sadasiva Rao, M. P. Pareek, and Rajat Gupta. A single chip pipelined cascadable multichannel signal processor. In *8th International Conference on VLSI Design*, pages 150–155. IEEE Computer Society, 1995.

[57] K. Kurosawa and T. Satoh. Generalization of higher order SAC to vector output Boolean functions. In *Advances in Cryptology - ASIACRYPT'96*, Lecture Notes in Computer Science, pages 218–231. Springer-Verlag, 1996.

[58] K. Kurosawa and T. Satoh. Design of SAC/PC(l) of order k Boolean functions and three other cryptographic criteria. In *Advances in Cryptology - EUROCRYPT'97*, Lecture Notes in Computer Science, pages 434–449. Springer-Verlag, 1997.

[59] P. Langevin. The covering radius of $R(1,9)$ into $R(3,9)$. In *Eurocode 90*, volume 514 of *Lecture Notes in Computer Science*, pages 51–59. Springer-Verlag, 1991.

[60] P. Langevin. On the orphans and covering radius of the Reed-Muller codes. volume 539 of *Lecture Notes in Computer Science*, pages 234–240. Springer-Verlag, 1991.

[61] F. J. MacWillams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North Holland, 1977.

[62] S. Maitra, B. K. Roy, and P. Sarkar. Ciphertext only attack on LFSR based encryption scheme. *Calcutta Statistical Association Bulletin*, 49(195-196):239–254, 1999.

[63] S. Maitra and P. Sarkar. Balancedness and correlation immunity of symmetric Boolean functions. *Communicated*.

[64] S. Maitra and P. Sarkar. Walsh transform and nonlinearity of symmetric Boolean functions. *Communicated*.

[65] S. Maitra and P. Sarkar. Enumeration of correlation immune Boolean functions. In *4th Australasian Conference on Information, Security and Privacy*, number 1587 in Lecture Notes in Computer Science, pages 12–25. Springer Verlag, April 1999.

[66] S. Maitra and P. Sarkar. Hamming weights of correlation immune Boolean functions. *Information Processing Letters*, 71(3-4):149–153, 1999.

[67] S. Maitra and P. Sarkar. Highly nonlinear resilient functions optimizing Siegenthaler's inequality. In *Advances in Cryptology - CRYPTO'99*, number 1666 in Lecture Notes in Computer Science, pages 198–215. Springer Verlag, August 1999.

[68] M. Morris Mano. *Digital Logic and Computer Design.* Prentice Hall (India), 1989.

[69] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, IT-15:122–127, January 1969.

[70] J. L. Massey. An introduction to contemporary cryptology. *Proceedings of the IEEE*, 76(5):533–549, May 1988.

[71] M. Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT'93*, Lecture Notes in Computer Science, pages 386–397. Springer-Verlag, 1994.

[72] S. E. Mcquillan and J. V. Mccanny. A systematic methodology for the design of high performance recursive digital filters. *IEEE Transactions on Computers*, 44(8):971–982, August 1995.

[73] W. Meier and O. Staffelbach. Fast correlation attack on stream ciphers. In *Advances in Cryptology - EUROCRYPT'88*, volume 330, pages 301–314. Springer-Verlag, May 1988.

[74] W. Meier and O. Staffelbach. Nonlinearity criteria for cryptographic functions. In *Advances in Cryptology - EUROCRYPT'89*, pages 549–562. Springer-Verlag, 1990.

[75] W. Meier and O. Stafflebach. Fast correlation attacks on certain stream ciphers. *Journal of Cryptology*, 1:159–176, 1989.

[76] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1997.

[77] W. Millan, A. Clark, and E. Dawson. An effective genetic algorithm for finding highly nonlinear Boolean functions. In *First International Conference on Information and Communications Security*, number 1334 in Lecture Notes in Computer Science, pages 149–158. Springer Verlag, 1997.

[78] W. Millan, A. Clark, and E. Dawson. Heuristic design of cryptographically strong balanced Boolean functions. In *Advances in Cryptology - EUROCRYPT'98.* Springer-Verlag, 1998.

[79] W. Millan, A. Clark, and E. Dawson. Boolean function design using hill climbing methods. In *4th Australasian Conference on Information, Security and Privacy*, number 1587 in Lecture Notes in Computer Science, pages 1–11. Springer Verlag, April 1999.

[80] C. J. Mitchell. Enumerating Boolean functions of cryptographic significance. *Journal of Cryptology*, 2(3):155–170, 1990.

[81] P. Sung Mo, L. Sangjin, S. Soo Hak, and K. Kwangjo. Improving bounds for the number of correlation immune Boolean functions. *Information Processing Letters*, 61(4):209–212, 1997.

[82] F. D. Murnaghan. The theory of group representations. *Baltimore*, 1938.

[83] J. J. Mykkeltveit. The covering radius of the (128, 8) Reed-Muller code is 56. *IEEE Transactions on Information Theory*, IT-26(3):358–362, 1983.

[84] S. Palit and B. K. Roy. Cryptanalysis of LFSR-encrypted codes with unknown combining functions. In *Advances in Cryptology - ASIACRYPT'99*, number 1716 in Lecture Notes in Computer Science, pages 306–320. Springer Verlag, November 1999.

[85] K. K. Parhi, C. Y. Wang, and A. P. Brown. Synthesis of control circuits in folded pipelined DSP architectures. *IEEE Journal of Solid State Circuits*, SC-27(1):29–43, January 1992.

[86] E. Pasalic and T. Johansson. Further results on the relation between nonlinearity and resiliency of Boolean functions. In *IMA Conference on Cryptography and Coding*, number 1746 in Lecture Notes in Computer Science, pages 35–45. Springer-Verlag, 1999.

[87] E. Pasalic, T. Johansson, S. Maitra, and P. Sarkar. New constructions of resilent and correlation immune boolean functions achieving upper bounds on nonlinearity. *Cryptology ePrint Archive, eprint.iacr.org, No. 2000/048*, 2000.

[88] 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.

[89] 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.

[90] W. Penzhorn. Correlation attacks on stream ciphers : Computing low weight parity checks based on error correcting codes. In *Fast Software Encryption, FSE'96*, volume 1039, pages 159–172. Springer-Verlag, 1996.

[91] N. Pippenger. Entropy and enumeration of Boolean functions. *IEEE Transactions on Information Theory*, 45(6):2096–2100, SEPTEMBER 1999.

[92] B. Preneel, R. Govaerts, and J. Vandewalle. Boolean functions satisfying higher order propagation criteria. In *Advances in Cryptology - EUROCRYPT'91*, Lecture Notes in Computer Science, pages 141–152. Springer-Verlag, 1991.

[93] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle. Propagation characteristics of Boolean functions. In *Advances in Cryptology - EUROCRYPT'90*, Lecture Notes in Computer Science, pages 161–173. Springer-Verlag, 1991.

[94] S. Ramanathan and V. Visvanathan. A systolic architecture for LMS adaptive filtering with minimal adaptation delay. In *9th International Conference on VLSI Design*, pages 286–289. IEEE Computer Society, 1996.

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

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

[97] R. A. Rueppel and O. J. Staffelbach. Products of linear recurring sequences with maximum complexity. *IEEE Transactions on Information Theory*, IT-33:124–131, January 1987.

[98] P. Sarkar and S. Maitra. Construction of nonlinear resilient Boolean functions. *Indian Statistical Institute, Technical Report No. ASD/99/30*, November 1999.

[99] P. Sarkar and S. Maitra. Highly nonlinear balanced Boolean functions with important cryptographic properties. *Indian Statistical Institute, Technical Report No. ASD/99/31*, November 1999.

[100] 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 491–512. Springer Verlag, 2000.

[101] P. Sarkar and S. Maitra. Nonlinearity bounds and constructions of resilient boolean functions. In *Advances in Cryptology - CRYPTO 2000*, number 1880 in Lecture Notes in Computer Science, pages 515–532. Springer Verlag, 2000.

[102] M. Schneider. *On the construction and upper bounds of balanced and correlation immune functions.* In *SAC'97*, January 1997.

[103] J. Seberry and X. M. Zhang. Highly nonlinear 0-1 balanced Boolean functions satisfying strict avalanche criterion. In *Advances in Cryptology, Auscrypt'92*, number 718 in Lecture Notes in Computer Science. Springer-Verlag, 1993.

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

[105] J. Seberry, X. M. Zhang, and Y. Zheng. On constructions and nonlinearity of correlation immune Boolean functions. In *Advances in Cryptology - EUROCRYPT'93*, pages 181–199. Springer-Verlag, 1994.

[106] J. Seberry, X. M. Zhang, and Y. Zheng. Structures of cryptographic functions with strong avalanche characteristics. In *Advances in Cryptology, Asiacrypt 94*, number 917 in Lecture Notes in Computer Science, pages 119–132. Springer-Verlag, 1995.

[107] W. Shan. The structure and the construction of correlation immune functions. *MS Thesis, NTE Institute, Xian*, 1987.

[108] M. Bhaskar Sherigar, A. S. Mahadevan, K. Senthil Kumar, and Sumam David. A pipelined paraller processor to implement MD4 message digest algorithm on Xilinx FPGA. In *11th International Conference on VLSI Design*, pages 394–399. IEEE Computer Society, 1998.

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

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

[111] D. Slepian. *On the number of symmetry types of Boolean functions on n variables.* *Canadian Journal of Mathematics*, 5:185–193, 1953.

[112] D. R. Stinson. Resilient functions and large sets of orthogonal arrays. *Congressus Numerantium*, 92:105–110, 1993.

[113] D. R. Stinson. *Cryptography , Theory and Practice.* CRC Press, 1995.

[114] D. R. Stinson and J. L. Massey. An infinite class of counterexamples to a conjecture concerning non-linear resilient functions. *Journal of Cryptology*, 8(3):167–173, 1995.

[115] Y. V. Tarannikov. On resilient Boolean functions with maximum possible nonlinearity. *Cryptology ePrint Archive, eprint.iacr.org, No. 2000/005*, 2000.

[116] A. F. Webster and S. E. Tavares. On the design of S-boxes. In *Advances in Cryptology - CRYPTO'85*, Lecture Notes in Computer Science, pages 523–534. Springer-Verlag, 1986.

[117] S. Wolfram. *The Mathematica Book, Mathematica Version 3, Third Edition*. Wolfram Media / Cambridge University Press, 1996.

[118] Y. X. Yang and B. Guo. Further enumerating Boolean functions of cryptographic significance. *Journal of Cryptology*, 8(3):115–122, 1995.

[119] A. M. Youssef, T. W. Cusick, P. Stanica, and S. E. Tavares. New bounds on the number of functions satisfying the strict avalanche criterion. In *SAC'96*, 1996.

[120] X. M. Zhang and Y. Zheng. Cryptographically resilient functions. *IEEE Transactions on Information Theory*, 43(5):1740–1747, 1997.