CONSTRUCTION OF CRYPTOGRAPHICALLY STRONG BOOLEAN FUNCTIONS WELL SUITED FOR SYMMETRIC CRYPTOSYSTEMS

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

BY

MANSOOR AHMED KHAN

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

AUGUST 2013

Approval of the thesis:

CONSTRUCTION OF CRYPTOGRAPHICALLY STRONG BOOLEAN FUNCTIONS WELL SUITED FOR SYMMETRIC CRYPTOSYSTEMS

submitted by **MANSOOR AHMED KHAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Cryptography, Middle East Technical University** by,

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

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

Prof. Dr. Ferruh Özbudak
Supervisor, **Department of Mathematics and Instituite of Applied Mathematics** _____

Examining Committee Members:

Prof. Dr. Ersan Akyıldız (Head of the examining com.)
Faculty of Arts and Sciences
Middle East Technical University _____

Prof. Dr. Ferruh Özbudak (Supervisor)
Department of Mathematics and Institute of Applied Mathematics, Middle East Technical University _____

Assoc. Prof. Dr. Ali Doğanaksoy
Department of Mathematics
Middle East Technical University _____

Assist. Prof Dr. Zülfükar Saygı
Department of Mathematics
TOBB University of Economics and Technology _____

Dr. Oğuz Yayla
Institute of Applied Mathematics
Middle East Technical University _____

Date: _____

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

Name, Last Name: MANSOOR AHMED KHAN

Signature :

ABSTRACT

CONSTRUCTION OF CRYPTOGRAPHICALLY STRONG BOOLEAN FUNCTIONS WELL SUITED FOR SYMMETRIC CRYPTOSYSTEMS

Ahmed Khan, Mansoor
PhD, Department of Cryptography
Supervisor : Prof. Dr. Ferruh Özbudak

August 2013, 51 pages

Boolean functions are amongst the vital ingredients of any symmetric cryptosystem in order to implement principles of confusion and diffusion. These are utilized as non-linear filtering functions or combiner functions in LFSR-based stream ciphers and as s-box component functions or non-linear encryption functions in Fiestel structure based block ciphers. Consequently, the cryptographic properties of Boolean functions are amongst the main contributors to the strength of these ciphers against cryptanalysis. The key cryptographic characteristics of Boolean functions include balanced-ness, non-linearity, correlation immunity and resilience, strict avalanche criteria and propagation criteria, and more recently, algebraic degree and algebraic immunity. Hence cryptographically strong Boolean functions are invariably required to posses superior cryptographic characteristics mentioned above in order to effectively resist all existing and potential cryptanalytic attack techniques.

The purpose of this research work is construction of cryptographically strong Boolean functions that can be utilized in symmetric cryptosystems offering effective resistance to existing cryptanalysis techniques. During the course of this research work, existing significant methods of construction would be studied and analyzed in depth. Based on this analysis, construction methods for Boolean functions with good cryptographic properties are aimed to be proposed. More focus would be directed to construction methods based on principles of finite fields and that involving combinatorial design theory. The significant constructions based on finite field principles include use of

primitive polynomials, primitive elements and block codes, while those based on combinatorial design theory depend on the use of combinatorial objects, such as relative difference sets, for constructing Perfectly Non-linear (PN) or Almost Perfectly Non-linear (APN) functions. In the end, the proposed constructions would be analyzed in terms of their cryptographic properties in comparison with other existing constructions in order to evaluate their efficacy for deployment in symmetric cryptosystems.

Keywords : Boolean Functions, Symmetric Cipher, Non-Linearity, Algebraic Immunity, Optimal Algebraic Immunity

ÖZ

SIMETRIK KRIPTOSISTEMLERDE KULLANİLABILECEK KRIPTOGRAFIK OLARAK GÜÇLÜ BOOLE FONKSIYONLARİNİN INŞA EDILMESIDIR

Ahmed Khan, Mansoor
Doktora, Kriptografi Bölümü
Tez Yöneticisi : Prof. Dr. Ferruh Özbudak

Ağustos 2013, 51 sayfa

Boole fonksiyonları, karmaşıklık ve yayılma prensiplerini uygulamaya çalışan herhangi bir simetrik kriptosistem için önem taşıyan yapılar arasındadır. LFSR tabanlı akış şifrelerinde doğrusal olmayan filtreleme fonksiyonları ya da birleştirici fonksiyonlar ile, Feistel yapılı blok şifrelerinde de S-kutuları ya da doğrusal olmayan şifreleme fonksiyonları ile bu prensipler sağlanmaktadır. Bu nedenle, Boole fonksiyonlarının kriptografik özellikleri, bahsedilen şifre sistemlerinin kriptoanalize dayanıklılığını sağlayan esas kriterler arasındadır. Boole fonksiyonlarının önemli kriptografik karakteristikleri dengeliliği, doğrusal olmamayı, korelasyon bağımlılığını ve dayanıklılığını, katı çığ etkisini, dağılma kriterini ve son olarak cebirsel derecesini ve cebirsel bağımlılığını içerir. Bu yüzden kriptografik olarak güçlü Boole fonksiyonlarının, bilinen ve potansiyel bütün kriptoanaliz saldırı tekniklerine karşı etkili olarak dayanmasını sağlamak için yukarıda bahsedilen üstün kriptografik özelliklere her şartta sahip olması gereklidir.

Bu araştırma çalışmasının amacı da simetrik kriptosistemlerde kullanılabilecek, bilinen kriptoanaliz tekniklerine etkili savunma sunan, kriptografik olarak güçlü Boole fonksiyonlarının inşa edilmesidir. Bu araştırma çalışması boyunca, bilinen güçlü inşa yöntemleri derinlemesine çalışılmış ve analiz edilmiştir. Bu analiz temel alınarak, iyi kriptografik özelliklere sahip Boole fonksiyonlarının inşa yöntemlerinin önerilmesi amaçlanmıştır. Sonlu cisimler cebiri prensiplerine dayalı, kombinatorik tasarım teorisinin de dahil olduğu inşa yöntemleri üzerinde daha çok durulmuştur. Sonlu cisimler ce-

birine dayalı inşalarda ilkel polinomlar, ilkel elemanlar ve blok kodları prensipleri kullanılmakta olup, kombinatorik tasarım teorisini kullananlarda ise Mükemmel Doğrusal olmayan (PN) ya da Neredeyse Mükemmel Doğrusal olmayan (APN) fonksiyonları inşa etmek için göreceli fark kümeleri gibi kombinatorik nesneler kullanılmıştır. Son olarak, önerilen inşalar kriptografik özellikleri açısından diğer bilinen inşalarla kıyaslanarak, simetrik sistemlerde kullanılabilirliğini değerlendirmek için analiz edilecektir.

Anahtar Kelimeler: Boole fonksiyonları, Simetrik Şifreler, Doğrusal Olmamayı, Cebirsel Bağımlılığını, Optimal Cebirsel Bağımlılığını

In the name of Allah, the most beneficent, the most merciful.
Dedicated to my parents, wife and children, their sincere prayers and ever-present support have been the driving forces for my success.

ACKNOWLEDGMENTS

I duly acknowledge all the help and support extended by consummate supervision of my advisor Prof. Dr. Ferruh Özbudak, which made it possible for me to complete my research work.

I would also like to acknowledge the thorough guidance and help extended to me by Dr. Oğuz Yayla, every time I needed it.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Cryptology

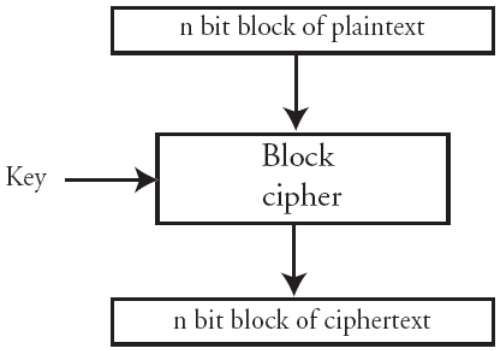
Cryptology has been used to protect information from unwanted disclosure for thousands of years. One of the earliest and commonly known examples is Caesar cipher employed by the Roman Emperor Julius Caesar around 60 BC for secret communications with his troops. But cryptology is not all about protecting information; the other aspect is trying to recover secret information from encrypted message using analytical or mathematical techniques. Naturally, cryptology has two main branches; Cryptography dealing with encrypting information for protection from disclosure to unwanted parties, and Cryptanalysis that deals with employing techniques to recover the original intelligible text from encrypted messages. Over the years, role of cryptology has been in constant evolution. While classical cryptology dealt primarily with making encryption schemes for encrypting data and breaking them by the adversaries to recover data, with the emergence of information age, new requirements such as user authentication, integrity of data and non-repudiation by users have been added to the applications of cryptology. In all applications, the original information is referred to as plaintext while the encrypted or coded messages are named ciphertext.

Cryptography can be further divided into two main categories; Asymmetric or Public Key cryptography and Symmetric or Private Key cryptography. In some literature, one way or hash functions are regarded as the third type of cryptographic schemes. The major difference between public and private key cryptography is that the earlier uses different keys for encryption and decryption while the latter utilizes same keys at both ends. Symmetric or private key cryptography consists of two main types of schemes; Block Ciphers and Stream Ciphers. Block ciphers usually employ a fixed encryption transformation on bigger blocks of the input message while stream ciphers use a variable one, commonly at single bit level.

Block ciphers are primarily based on the principles of Confusion and Diffusion introduced by Claude Shannon in [34]. Confusion is transforming the relationship between secret key and ciphertext as complicated as possible. Diffusion deals with propagating the change in a single bit of plaintext over the complete ciphertext bits. Non-linear transformations are used to create confusion while diffusion is achieved by linear transformations. The transformations are usually applied recursively and repeatedly

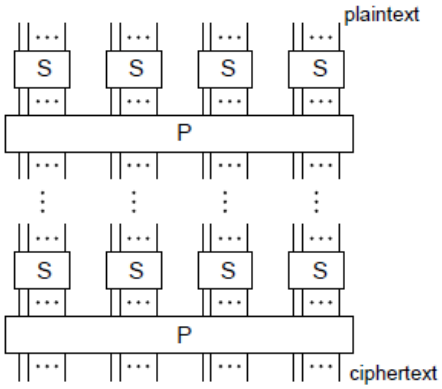
to achieve both confusion and diffusion. Each repetition is called a round and uses a different round key, derived from the secret key. A standard block cipher design is shown in Figure 1.1.

Figure 1.1: A typical block cipher

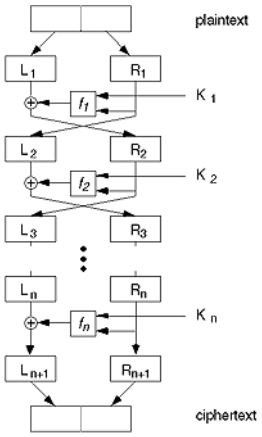


There are two main structures used in block cipher design, Feistel structure and Substitution Permutation Network (SPN). Feistel structure employs a defined transformation on half of the input block in each round while the SPN round modifies the complete input block using the round keys. An example of SPN and Feistel structure is shown in Figure 1.2. Boolean functions find their application in designing these round functions for block ciphers.

Figure 1.2: SPN and Feistel structures



(a) SPN structure

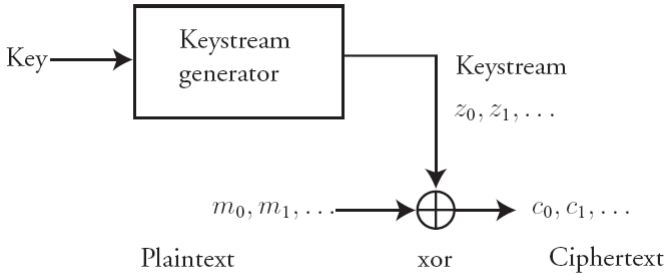


(b) Fiestel structure

Unlike block ciphers, stream ciphers commonly use a simple function, the Exclusive OR (XOR), between plaintext bits and the key bits to obtain ciphertext. This, however, mandates the length of secret key to be either as long as the plaintext or sufficiently long to resist cryptanalysis. The Vernam cipher or One Time Pad (OTP) [41] is a stream cipher that uses a random bit stream as secret key XORed with the plaintext to produce the cipher text. The length of plaintext and the secret key is equal and as the name suggests, each key is used only once. It is arguably the lone provably secure

cryptosystem but is quite impractical due to required length of the secret key and no re-use restriction. The modern stream ciphers employ a smaller psuedo random bit stream as secret key in comparison and try to imitate the concept of OTP. A typical stream cipher structure is shown in Figure 1.3.

Figure 1.3: A typical stream cipher



Linear Feedback Shift Registers (LFSRs) are mostly used to generate the pseudo random bit streams in stream ciphers. Owing to the Berlekamp-Massey algorithm [18], use of merely LFSRs to generate key streams is not enough and some non-linearity needs to be induced to increase the Linear Complexity (LC) of the generated key stream. This is achieved by employing Non-Linear Filter Generator (Figure 1.4) and/or Non-Linear Combiner designs (Figure 1.5) in stream ciphers. Once again, Boolean functions are integral part of both these structures.

Figure 1.4: Non-linear filter generator

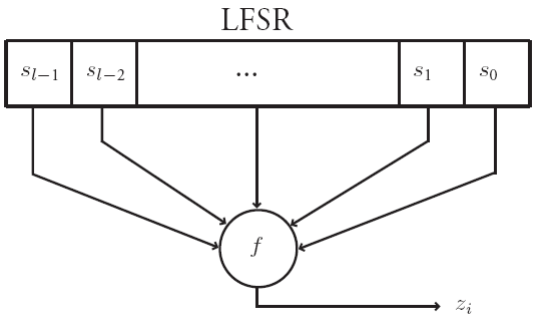
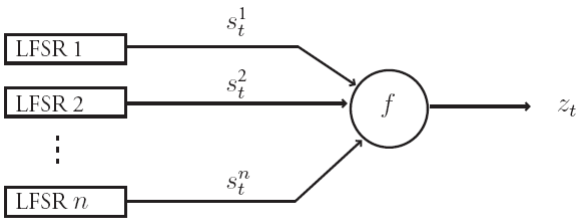


Figure 1.5: Non-linear combiner



1.2 Motivation for research

As evident from discussion in the preceding section, Boolean functions play an imperative role in design of almost every modern symmetric cipher. They are either utilized as non-linear filtering functions or combiner functions in LFSR-based stream ciphers or as S-Box component functions and non-linear encryption functions in Feistel structure based block ciphers. Resultantly, the cryptographic properties of Boolean functions become the primary contributors to the strength of these ciphers against cryptanalysis. The important cryptographic characteristics of Boolean functions include Balancedness, high Non-Linearity, Correlation Immunity and Resiliency, Strict Avalanche Criteria and Propagation Criteria, and more recently, high Algebraic Degree and optimal Algebraic Immunity. It, therefore, becomes needless to say that Boolean functions are required to invariably possess strong cryptographic characteristics in order to adequately resist all existing and potential cryptanalytic attack techniques.

In [13, 14], N. Courtois, and W. Meier presented Algebraic and Fast Algebraic attacks on stream ciphers with linear feedback. Subsequently, some variants of these attacks were devised to further improve their efficiency [1–3, 26, 30]. This triggered a series of research work in which several constructions of Boolean functions were proposed focused on attaining high algebraic degree and optimal or sub-optimal algebraic immunity, while maintaining high non-linearity [5, 8–11, 15, 16, 28, 39, 40, 42, 43, 45]. In [11], C. Carlet and K. Feng proposed an infinite class of Boolean functions that possessed balanced-ness, high algebraic degree, optimal algebraic immunity, high non-linearity compared and good immunity to fast algebraic attacks. The proposed construction in [11] is based on selecting a primitive element $\alpha \epsilon \mathbb{F}_2^n$ and selecting its consecutive powers from 1 to $(2^{n-1} - 2)$, along with "0" and "1" vector in the support set of the function. Subsequently in [45], X.Zeng, C.Carlet, J.Shan, L.Hu presented three more constructions, achieving either the same or in some cases, even higher non-linearities, while maintaining the degree and algebraic immunity as in [11]. These construction methods also utilized a primitive element $\alpha \epsilon \mathbb{F}_2^n$ and selecting its powers in the support set of the function. However, the powers selected in this case were not consecutive, rather based on some pre-defined sets. Although the infinite class achieved very high values of non-linearity, they were not close to bent functions, thus leaving room for further improvement in non-linearity, while preserving the rest of the properties.

In [40], Z. Tu, and Y. Deng presented a combinatorial conjecture and constructed a new class of balanced Boolean functions combining ideas in [11, 17, 19]. The functions belonging to this new class also attained maximal algebraic degree for balanced functions, optimal algebraic immunity and non-linearity better than [11, 45]. However, in [12], C. Carlet pointed out a weakness in the construction. The product of constructed functions with any Linear function reduced the degree of the resultant function by almost half, making it vulnerable to fast algebraic attacks [1–3, 13, 14, 26, 30]. A repair was also suggested in the same work to remove this weakness but the rest of the properties including algebraic degree and resistance to fast algebraic attacks were being studied. A modified family of functions in [40] was presented in [37] also by X. Tang, D. Tang, X. Zeng, L. Hu, but the vulnerability to the weakness of original construction

and resistance to algebraic and fast algebraic attacks was not investigated. Therefore, construction of an infinite class of functions with comparable cryptographic properties, while also offering good resistance to algebraic and fast algebraic attacks was still an open area.

1.3 This Thesis

The constructions proposed in [11] and [45] clearly demonstrated, and also proved mathematically, that selecting different powers of the primitive elements affected the non-linearity of the functions, along with their algebraic degree and algebraic immunity. In this research work, we have devised two algorithms by modifying the genetic hill climbing algorithm [40] for improvement in the non-linearity of functions constructed using the infinite class proposed in [11] for number of variables $n \geq 8$. The improved functions not only possess higher non-linearity than the original functions in [11], but also maintain the high algebraic degree and optimal algebraic immunity. The improvement algorithms have been verified by constructing all Boolean functions for $8 \leq n \leq 11$ and improving their non-linearity. Chapter 3 covers the details of this research work.

The second contribution in this thesis is construction of two hybrid classes of balanced Boolean functions based on ideas in [11, 17, 19] by modifying the construction in [40]. The modified functions not only maintain their cryptographic properties i.e. balancedness, maximal algebraic degree for balanced functions, optimal algebraic immunity and very high non-linearity, but also avoid the weakness pointed out in [12]. We also practically analyse and verify (using MAGMA) that the functions constructed in the two proposed hybrid classes are not comparably vulnerable to fast algebraic attacks as functions in [40]. Details of this contribution are discussed in Chapter 4.

Finally, a 1-resilient class of Boolean functions with high algebraic degree, optimal algebraic immunity and high non-linearity has also been proposed by using ideas in [37, 39] and modifying our second construction in Chapter 4. This construction was also implemented in MAGMA and analysed to verify their cryptographic properties. An analysis for resistance against algebraic and fast algebraic attacks was also performed which shows that functions belonging to this class offer good resistance to these attacks. Chapter 5 includes the details of the said construction.

CHAPTER 2

BOOLEAN FUNCTION BASICS

Let us first discuss the basics of Boolean functions relevant to this research work. We start off with some definitions. Let $\mathbb{F}_2$ define the Binary Field. Then $\mathbb{F}_2^n$ can be visualized as an n-dimensional vector space over $\mathbb{F}_2$. A Boolean function f on n-variables can be envisaged as a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Let $\mathfrak{B}_n$ denote the set of all Boolean Functions from $\mathbb{F}_2^n$ into $\mathbb{F}_2$.

2.1 Boolean function representations

2.1.1 Truth Table and Sequence

Any Boolean function $f(x_1, ..., x_n)$ can be represented as a binary string of length 2^n with each representing the output of the function with respect to the ordered pair $(x_1, ..., x_n)$ as the input:-

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

This is known as the Truth Table of f. The Sequence of f denoted by Seq(f) is a $(1, -1)$ valued mapping of the truth table obtained by $Seq(f) = 1 - 2f = (-1)^f$.

2.1.2 Algebraic Normal Form (ANF) and Algebraic Degree

Any n-variable Boolean function can be considered as a multivariate polynomial over $\mathbb{F}_2$. This polynomial can be represented as a sum of distinct variables, each of order k while $0 \leq k \leq n$. Then

$$f(x_1, x_2, ..., x_n) = a_0 + \left(\sum_{1 \leq i \leq n} a_i x_i\right) + ... + \left(\sum_{1 \leq i < j \leq n} a_{i,j} x_i x_j\right) + ... + (a_{1,2,...,n} x_1 x_2 ... x_n) \tag{2.2}$$

where each $a_i \epsilon \mathbb{F}_2$. The above representation is called the Algebraic Normal Form (ANF) of f. The maximum number of variable x appearing in a single term amongst all terms of the ANF is called the Algebraic Degree of f.

2.1.3 Trace Representation

Recall that the trace function from $\mathbb{F}_2^n$ to it's sub-field $\mathbb{F}_2^k$ is defined as

$$tr_k^n(x) = x + x^q + x^{q^2} + ... + x^{q^d-1}$$

where $q = 2^k$ and $d = n/k$. Therefore if *p(x)* is a polynomial of degree $\leq 2^n - 1$ and $x \epsilon \mathbb{F}_2^n$, then the Boolean function f can also be represented by *tr(p(x))*, called it's trace representation.

2.2 Weight and Support

The weight of a Boolean function wt(f), sometimes also referred to as the Hamming Weight, is the number of 1s in its truth table representation. The Support of f , Supp(f) is defined as

$$supp(f) = \{\forall\, x \mid f(x) = 1\}\,. \tag{2.3}$$

2.3 Balanced-ness

An n-variable Boolean function is called Balanced if $wt\,(f) = 2^{(n-1)}$, i.e its support set supp(f) has dimension $2^{(n-1)}$.

2.4 Walsh Spectrum

For $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ and $\omega = (\omega_1, \omega_2, ..., \omega_n)$, define $\alpha \cdot \omega$ as the usual inner product $\alpha \cdot \omega = (\alpha_1\omega_1, \alpha_2\omega_2, ..., \alpha_n\omega_n)$. Then the Wash transform of f, W_f is calculated as

$$W_f(\alpha) = \sum_{\alpha \epsilon \mathbb{F}_2^n} (-1)^{f(\alpha)+\alpha.\omega}. \tag{2.4}$$

Obviously, each coefficient in the Walsh spectrum has values between 2^n and -2^n. Note that for a balanced Boolean function, $W_f(0) = 0$. The total energy in the Walsh spectrum is conserved, as established in Parseval's Identity

$$\sum_{\alpha \epsilon \mathbb{F}_2^n} W_f^2(\alpha) = 2^{2n}. \tag{2.5}$$

2.5 Non-Linearity

The Non-Linearity of f, nl(f) is given by

$$nl(f) = 2^{(n-1)} - \frac{1}{2} \max_{\alpha \epsilon \mathbb{F}_2^n} \left| W_f(\alpha) \right|. \tag{2.6}$$

2.6 Bent Function

A Boolean function f in $\mathbb{F}_2^n$ is called "Bent" if it's Walsh spectrum is two valued, i.e. $W_f(\alpha) = \pm 2^{n/2} \; \forall \, \alpha \, \epsilon \, \mathbb{F}_2^n$, where n is always even. Clearly, a bent function is unbalanced since $W_f(0) \neq 0$.

2.7 Correlation Immunity and Resilience

Correlation Immunity (CI) = k implies that the output of the function is statistically independent of the combination of any k of its inputs. In terms of the Walsh spectrum, $W_f(\alpha) = 0$ for all α with $1 \leq wt(\alpha) \leq k$. A Boolean function has Resiliency = k if it is balanced and has correlation immunity = k. In other words, a k-resilient function has $W_f(\alpha) = 0$ for all α with $0 \leq wt(\alpha) \leq k$.

2.8 Annihilator of a Function

The Annihilator of f , AN(f) is a Boolean function g such that $f * g = 0$, where $f * g$ is the usual product of functions $f * g = f(x).g(x)$.

2.9 Algebraic Immunity

This brings us to the last definition that is Algebraic Immunity of f, AI(f) which is determined as the minimum degree non-zero annihilator of f

$$AI(f) = \min \{ \, deg(g) \; | \; \forall \, g \, \epsilon \, \mathfrak{B}_n \; st \; f(x).g(x) \; = \; 0, \; \forall \, x \, \epsilon \, \mathbb{F}_2^n \}. \tag{2.7}$$

High value of algebraic immunity insinuates that non-linearity of the function is fairly high. This was established in [25] and is now commonly known as the Lobanov's Bound for minimum value of non-linearity given it's algebraic immunity.

$$nl(f) \geq 2 \sum_{0 \leq i \leq (AI(f)-2)} \binom{n-1}{i}. \tag{2.8}$$

2.10 Some Cryptanalytic Attacks

2.10.1 Berlekamp-Massey Attack

Definition 2.1. Linear Complexity (LC) of a periodic bit sequence is defined as the length of the shortest LFSR that can produce this sequence. A bit sequence S_t is periodic when it repeats after a finite number of bits i.e. for an integer $p \geq 0, S_t = S_{t+p} \forall t \geq 0$. This integer p is called the period of the recurring sequence. Linear Complexity of a sequence is ∞ when it can not be produced by an LFSR.

Berlekamp-Massey attack is based on the Berlekamp-Massey algorithm proposed in [18] to calculate the Linear Complexity of a pseudo-random bit sequence. It also recovers the connection polynomial and size of an LFSR that can be equivalently used to generate the sequence being analysed. It requires $\geq 2.LC$ number of bits of a sequence to construct an LFSR of length $L = LC$ and evaluate it's connection polynomial that generates the sequence.

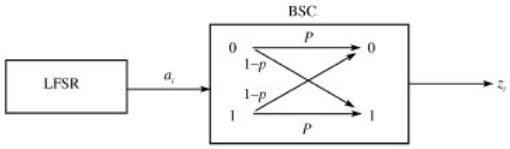
2.10.2 Correlation and Fast Correlation Attacks

Correlation attack [25] targets the improperly selected combining function f in non-linear combiner model (Figure 1.5). If the key-stream obtained as output of this function is correlated to one of the input LFSRs more than others, a divide-and-conquer approach can be employed to recover the initial state of each LFSR separately. This reduces the attack complexity from Brute Force search = $\prod_{1 \leq i \leq k} IS_i$ to $\sum_{1 \leq i \leq k} IS_i$, where IS_i stands for all possible combinations of initial states for the i^{th} LFSR and k is the total number of LFSRs used.

Since it is possible to utilise a divide-and-conquer approach on the non-linear combiner model, Fast Correlation attacks [27] envisage to recover the output of a single LFSR with known connection polynomial as a decoding problem based on observed key-stream. Their attack set-up is shown in Figure 2.1.

As shown in the Figure 2.1, output of the register is assumed to be filtered by a Binary Symmetric Source that is memory-less. This action combines output of registers in the design as well as the combining function over the key-stream output. Hence, problem

Figure 2.1: Fast Correlation attack set-up



of recovering the output of this register is equivalent to decoding a message transmitted over a noisy channel and the correlation of the key-stream to one or more of the input register becomes the probability of each output bit being/ not being flipped. Meier and Staffelbach devised two algorithms to decode the output key-stream from the noisy one using known structure of the register(including it's connection polynomial). This attack suffered a huge limitation on length of the LFSRs used to a maximum of 10 stages to be successful. However, it was intriguing that a cryptographic problem could be envisaged as a decoding problem and the attack was modified to make it more efficient [6].

2.10.3 Algebraic and Fast Algebraic Attacks

Algebraic attacks [13] recover the initial states of an LFSR by simultaneous solution of a system of non-linear equations. These equations are constructed by observing the output key-stream after a sufficient number of output bits are generated in relation to the combiner/ non-linear filter function used and connection polynomial of the LFSR.

$$\begin{aligned} f(IS) &= s_0 \\ f(IS+1) &= s_1 \\ f(IS+2) &= s_2 \\ &\vdots \end{aligned}$$

where f is the combiner/ non-linear filter function, IS is the initial state of LFSR and $(IS + 1) = CP(IS)$, CP being the connection polynomial of LFSR. Since the connection polynomial of the LFSR is a linear function, degree of each of the equations in constructed system is equal to the degree of combining function. This degree, can however be reduced further either by using a function g such that $(f \oplus 1).g = 0$ or a function h such that $f.h = 0$ [26]. After obtaining the system of linear equations, it is solved using methods such as Linearisation or Gröbner Basis [20–22].

In fast algebraic attack [14], the degree of non-linear equations to be solved simultaneously is attempted to be further reduced by looking for relations between the initial state of the LFSR and more output bits, compared to a single output bit in each step in case of algebraic attack. This makes the attack more efficient. The attack is dependant solely on the existence of functions g and h with degrees (e, d), $e < d$ respectively,

such that $f.g = h$ further improves the efficiency of the attack. However, it requires additional pre-computations to calculate linear combinations, required to cater for considering several output key-stream bits instead of one for each equation in the system.

2.11 Resisting Cryptanalytic Attacks

As mentioned earlier, a Boolean function to be utilized in a symmetric cipher must be balanced, possess high algebraic degree, high non-linearity and optimal algebraic immunity. A high algebraic degree enables the function to resist Berlekamp-Massey attack [18], high non-linearity contributes to enduring fast correlation attacks [6, 27], while high algebraic immunity is a necessary but not sufficient condition to counter algebraic and fast algebraic attacks [1–3, 13, 14, 26, 30]. The optimal value of algebraic immunity is $\lceil \frac{n}{2} \rceil$. It is elaborated in [12] that if we can find g of low algebraic degree and $h \neq 0$ of feasible algebraic degree such that $f * g = h$, then the function f becomes vulnerable to fast algebraic attack. To attain optimal resistance to fast algebraic attacks for an n-variable function f , there should not exist two functions $g \neq 0$ and h such that $f * g = h$ and $deg(g) + deg(h) < n$ while $deg(g) < n/2$. The function f is the "weakest" when there exists a function g of degree 1 (linear function), and a function h with degree $\lceil \frac{n}{2} \rceil$ such that $f * g = h$. The "next to weakest" case is when there exists a function g of degree 1 (linear function), and a function h with degree $\lceil \frac{n}{2} \rceil + 1$ with $f * g = h$.

CHAPTER 3

IMPROVEMENT IN NON-LINEARITY OF CARLET-FENG INFINITE CLASS

3.1 The Carlet-Feng Infinite Class of Boolean Functions

We now describe the infinite class of balanced Boolean functions with high algebraic degree, optimal algebraic immunity, good immunity to fast algebraic attacks and good non-linearity as proposed in [11] by C. Carlet and K. Feng. Let $\alpha \epsilon \mathbb{F}_2^n$ be the primitive element of F_2^n. Then Boolean function f from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ for number of variables n whose support set is $supp(f) = \left\{0, 1, \alpha, \alpha^2, ..., \alpha^{2^{(n-1)}-2}\right\}$ has optimal algebraic immunity i.e.$\lceil \frac{n}{2} \rceil$. The algebraic degree of f is $(n-1)$ and it is balanced. Furthermore the non-linearity of f is given by

$$nl(f) \geq 2^{(n-1)} + \frac{2^{\frac{n}{2}}+1}{\pi} \ln\left(\frac{\pi}{4(2^n-1)}\right) - 1 \approx 2^{(n-1)} - \frac{2\ln 2}{\pi} n 2^{\frac{n}{2}}. \tag{3.1}$$

Mathematical proofs of the above relations are presented in [11] and we do not reproduce them here. It is mentioned that these functions, owing to their high algebraic degree, optimal algebraic immunity and good non-linearity, behave well against fast correlation attacks [17, 19], algebraic attacks, fast algebraic attacks [1–3, 13, 14, 26, 30] and Berlekamp-Massey attack [12].

Specifically in connection with resistance to fast algebraic attacks, computer investigations were performed to discover that for the proposed class of functions, no non-zero function g of degree at most e and no function h of degree at most d exists such that $f * g = h$ when $(e, d) = (1, n-2)$ for odd n and $(e, d) = (1, n-3)$ for even n; both verified when $n \leq 12$. In case of $e > 1$, the functions g and h with respective degrees (e, d) such that $(e+d) < (n-1)$ were not observed for $n \leq 9$ and $e < n/2$, for $n = 10$ and $e \leq 3$ and for $n = 11$ and $e \leq 2$. It is also highlighted that before this construction [11], no infinite class of Boolean functions with high algebraic degree, good algebraic immunity and good non-linearity was presented.

In [45], three more classes of Boolean functions were proposed by X.Zeng, C.Carlet, J.Shan, L.Hu. In this case powers of the primitive element $\alpha \epsilon \mathbb{F}_2^n$ to be included in support of the function f were chosen based on some pre-defined sets. While the

method improved non-linearity of some constructed functions, the rest had the same non-linearity as in [11]. Hence these constructions did not guarantee a higher non-linearity than [11] for all functions. Later in [28, 39, 42, 43] more constructions were presented to achieve optimal algebraic immunity, good non-linearity and in some cases 1-resiliency. However, none of these attained significant increase in the non-linearity of functions as compared to [11].

Since construction in [45] employs a different criterion for selection of powers of the primitive element $\alpha \epsilon \mathbb{F}_2^n$ than [11] and achieves some functions with better non-linearity, we studied the behaviour of functions in detail. While [11] presents an easy selection criteria, the one used in [45] is comparatively intricate. Subsequently, our focus was directed to using the construction in [11] as the starting point and then changing the powers of α in the support set based on the affect on Walsh spectrum of the functions. This led to the development of a relatively simple algorithm derived from the hill climbing approach [40], based on the behaviour of Walsh spectrum of Boolean functions. The algorithm improves in non-linearity of functions while preserving algebraic degree and algebraic immunity. Non-linearity of most functions constructed by [11] was improved using this algorithm, while some could not be improved. Subsequently, a second algorithm was developed by modifying the first one to improve non-linearity of remaining functions as well. Therefore as compared to [45], which improves non-linearity in case of some functions as compared to [11], we achieve better non-linearity for all functions.

3.2 Algorithms for Improving Non-linearity

3.2.1 The Behaviour of Walsh Spectrum

Before we describe the algorithms developed, we first review the behaviour of the Walsh spectrum of a Boolean function and the effects caused by changes in the truth table of a function. Recall that the Sylvester-Hadamard matrix, also known as the Walsh-Hadamard matrix, is defined as follows

$$H_0 = 1,\ H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_n = H_{n-1} \otimes H_1 = H_{n-1} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}. \tag{3.2}$$

The symbol $\otimes$ denotes the usual Kronecker product. It is clear that H_n is a matrix of order 2^n. Using this matrix, the Walsh Transform of a function, also called the Walsh-Hadamard transform, can be easily calculated [23]. Given the sequence of a Boolean function $Seq(f) = (y_0, y_1, ..., y_{2^n-1})$, the Walsh spectrum can be computed as

$$H_f = H_n x[y_0, y_1, ..., y_{2^n-1}] = H_n[y_0, y_1, ..., y_{2^n-1}]^T$$

$$= H_n \begin{bmatrix} y_0 \\ y_1 \\ . \\ . \\ . \\ y_{2^n-1} \end{bmatrix} = \begin{bmatrix} A & + & B \\ A & - & B \end{bmatrix} \tag{3.3}$$

$$where\ A = \begin{bmatrix} y_0 \\ y_1 \\ . \\ . \\ . \\ y_{2^{n-1}-1} \end{bmatrix} and\ B = \begin{bmatrix} y_{2^{n-1}} \\ y_{2^n} \\ . \\ . \\ . \\ y_{2^n-1} \end{bmatrix}.$$

Hence the Walsh spectrum can be calculated recursively by using Equation 3.2. Let us demonstrate the process by an example

Example 3.1. Let f be a 3-variable Boolean function with the truth table $f(x_1, ..., x_3) = (0, 1, 1, 1, 1, 0, 0, 0)^T$. Using the recursion described in Equation 3.2, the Walsh spectrum computation can be performed as follows

$$Seq(f) = (1, -1, -1, -1, -1, 1, 1, 1)^T$$
$$W_f = H_3 \begin{bmatrix} 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix}^T$$

$$W_f = H_3 \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} A & + & B \\ A & - & B \end{bmatrix} = \begin{bmatrix} H_2 \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} & + & H_2 \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\ H_2 \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} & - & H_2 \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} H_1\begin{bmatrix} 1 \\ -1 \end{bmatrix} + H_1\begin{bmatrix} -1 \\ -1 \end{bmatrix} \\ H_1\begin{bmatrix} 1 \\ -1 \end{bmatrix} - H_1\begin{bmatrix} -1 \\ -1 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} H_1\begin{bmatrix} -1 \\ 1 \end{bmatrix} + H_1\begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ H_1\begin{bmatrix} -1 \\ 1 \end{bmatrix} - H_1\begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{bmatrix} \\ \begin{bmatrix} H_1\begin{bmatrix} 1 \\ -1 \end{bmatrix} + H_1\begin{bmatrix} -1 \\ -1 \end{bmatrix} \\ H_1\begin{bmatrix} 1 \\ -1 \end{bmatrix} - H_1\begin{bmatrix} -1 \\ -1 \end{bmatrix} \end{bmatrix} - \begin{bmatrix} H_1\begin{bmatrix} -1 \\ 1 \end{bmatrix} + H_1\begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ H_1\begin{bmatrix} -1 \\ 1 \end{bmatrix} - H_1\begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} + \begin{bmatrix} -2 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 2 \end{bmatrix} - \begin{bmatrix} -2 \\ 0 \end{bmatrix} \end{bmatrix} + \begin{bmatrix} \begin{bmatrix} 0 \\ -2 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ -2 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} \end{bmatrix} \\ \begin{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} + \begin{bmatrix} -2 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 2 \end{bmatrix} - \begin{bmatrix} -2 \\ 0 \end{bmatrix} \end{bmatrix} - \begin{bmatrix} \begin{bmatrix} 0 \\ -2 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 \\ -2 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} -2 \\ 2 \\ 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ -2 \\ -2 \\ -2 \end{bmatrix} \\ \begin{bmatrix} -2 \\ 2 \\ 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 2 \\ -2 \\ -2 \\ -2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -4 \\ 4 \\ 4 \\ 4 \end{bmatrix}.$$

Now, we observe that a single bit change in the truth table of the function f changes the Seq(f) either from 1 to -1 or vice versa. Hence the Walsh spectrum values will either be unaffected or would increase/ decrease by a value of 2. This affect is independent to the number of values since for larger variables, only the number of Walsh spectrum values changed would differ, but the deviation would always be ± 2. Hence according to Equation 2.6, the non-linearity of the function is increased or decreased by a value 1 with a single bit change in the truth table. Therefore, if suitable element of support set of the function is interchanged with the set of roots, that is, two suitable values in the truth table are swapped; the maximum coefficients in the Walsh spectrum is reduced by 4. Resultantly, the non-linearity of the function can be increased by a value of 2.

3.2.2 Algorithm 1

Let us fix some notations before presenting the algorithms. The array TT() holds the truth table of the Boolean Function constructed using [11]. ITT() holds the truth table of the improved function. Walsh() refers to the routine that calculates the Walsh spectrum of the Boolean function from its truth table representation. The algorithm is presented below

ALGORITHM 1

Walsh(TT())
maxWalsh = max | $Walsh(TT())$ |
copy TT() → ITT()
LastCount = 0
ReRun:
Count = LastCount + 1

if $Count < (2^n - 1)$ **then**
  for $i = \alpha^{count} \rightarrow \alpha^{2^{(n-1)}-2}$ **do**
    ITT(i) = 0
    LastCount = i
    Walsh(ITT())
    maxWalsh2 = max | $Walsh(ITT())$ |
    if $maxWalsh2 < maxWalsh$ **then**
      exit For
    else
      ITT(i) = TT(i)
    end if
  end for
  if $maxWalsh2 \geq maxWalsh$ **then**
    then GoTo Skipj:
  end if
  for $j = \alpha^{2^{(n-1)}-1} \rightarrow \alpha^{2^n-2}$ **do**
    ITT(j) = 1
    Walsh(ITT())
    maxWalsh3 = max | $Walsh(ITT())$ |
    if $maxWalsh3 < maxWalsh2$ **then**
      then exit For
    else
      ITT(j) = TT(j)
    end if
  end for
  if $maxWalsh3 = maxWalsh - 4$ **then**
    then output "Function Improved"
  else
    GoTo ReRun:
  end if
end if

Skipj:
if $maxWalsh2 \geq maxWalsh$ **or** $maxWalsh3 > maxWalsh - 4$ **then**
  output "Function could not be improved" and exit
end if.

In contrast to the generic hill climbing approach, Algorithm 1 changes an element of the support set to a root and determines its suitability based on change in the Walsh spectrum instead of searching all possible pair swaps. If the maximum Walsh value is reduced, it keeps this change and looks for a suitable change of a root to the support set. Once the suitable root to support swap is found, the maximum Walsh value is reduced by 4 and the non-linearity of the function improves by 2. This reduces the number of steps since the swap occurs only if the first change is suitable to increase the non-linearity. The support to root swap is handled in the loop for variable "i" and the loop for "j" handles the root to support swap to complete the improvement in non-linearity. In case non-linearity improvement is not achieved once a support to root swap is selected and all possible root to support interchanges have been tried, algorithm is re-run by incrementing the "LastCount" variable within the for loop for "i". This ensures the next suitable support to root swap is selected.

3.2.3 Algorithm 2

Algorithm 2 is an iterative application of Algorithm 1. However, it's also different to Algorithm 1 in the sense that it accepts a change in truth table even if the maximum value of Walsh spectrum is not decreased, but the number of maximum Walsh values is decreased in intermediate steps. Maximum Walsh value is ultimately decreased in the final iteration (non-linearity of functions was improved in at the most 4 iterations in all cases). Hence, increase in non-linearity of function by 2 is achieved in a similar manner as explained in case of Algorithm 1 in the preceding paragraph, except addition of variable "Count2" that determines the total number of swaps/ iterations allowed. The algorithm is presented below

---

ALGORITHM 2

Walsh(TT())
maxWalsh = max | $Walsh(TT())$ |
nmaxWalsh = # maxWalsh
copy TT() → ITT()
LastCount = 0
Count2 = 1

ReRun:
Count = LastCount + 1
if $Count < (2^n - 1)$ **then**
  while $Count2 \leq 6$ **do**
    for $i = \alpha^{count} \rightarrow \alpha^{2^{(n-1)}-2}$ **do**
      ITT(i) = 0
      LastCount = i
      Walsh(ITT())
      maxWalsh2 = max | $Walsh(ITT())$ |

nmaxWalsh2 = # maxWalsh2
if $maxWalsh2 < maxWalsh$ **or** $nmaxWalsh2 < nmaxWalsh$ **then**
Count2 +=1 and exit For
else
ITT(i) = TT(i)
end if
end for
if $maxWalsh2 \geq maxWalsh$ **then**
GoTo Skipj:
end if
for $j = \alpha^{2^{(n-1)}-1} \rightarrow \alpha^{2^n-2}$ **do**
ITT(j) = 1
Walsh(ITT())
maxWalsh3 = max | $Walsh(ITT())$ |
nmaxWalsh3 = # maxWalsh3
if $maxWalsh3 < maxWalsh2$ **or** $nmaxWalsh3 < nmaxWalsh2$ **then**
Count2 +=1 and exit For
else
ITT(j) = TT(j)
end if
end for
if $maxWalsh3 = maxWalsh - 4$ **then**
output "Function Improved"
else
Count2 -=1 and GoTo ReRun:
end if
end while
end if

Skipj:
if $maxWalsh2 \geq maxWalsh$ **or** $maxWalsh3 > maxWalsh - 4$ **then**
output "Function could not be improved" and exit
end if.

3.3 Advantages obtained

By employing Algorithms 1 and 2, the non-linearity of all the functions constructed using Carlet-Feng infinite class of Boolean functions [11] can be improved by at least 2 for number of variables $n \geq 8$. Additionally the algorithms preserve the balanced-ness, maximal algebraic degree and optimal algebraic immunity of the functions. Table 3.1 and 3.2 list some selected results for $n = 8$ to demonstrate the improvement in non-linearity for Algorithm 1 and 2 respectively, although all defining polynomials and primitive elements have been practically verified for $8 \leq n \leq 11$. Some results for $n = 9$, 10 and 11 are included in Appendix A, B and C respectively for reference.

Another significant advantage of the method is that the swapping of support set element

with the root resulting in improvement in non-linearity of the functions is not unique, i.e more than one such pairs exist. Resultantly, whilst there exists only one function for a fixed defining polynomial and a fixed primitive element in the infinite class [11], more than one function with higher non-linearity can be obtained using algorithm 1 or 2 by just changing the value of variable "LastCount". Same is demonstrated by some examples presented in Table 3.3 for $n = 8$, although it has been practically verified for all values.

Table 3.1: Comparison of non-linearities for $n = 8$ by Algorithm 1

Defining Polynomial (Integer value)	**Primitive element (Integer value)**	**Non-linearity of function in [11]**	**Elements swapped (root↔support)**	**Non-linearity of improved function**
285	2	112	$\alpha^{104} \leftrightarrow \alpha^{230}$	114
299	128	112	$\alpha^{66} \leftrightarrow \alpha^{147}$	114
301	57	112	$\alpha^{101} \leftrightarrow \alpha^{233}$	114
333	16	112	$\alpha^{87} \leftrightarrow \alpha^{241}$	114
351	4	112	$\alpha^{1} \leftrightarrow \alpha^{238}$	114
355	26	112	$\alpha^{94} \leftrightarrow \alpha^{221}$	114
357	101	112	$\alpha^{74} \leftrightarrow \alpha^{145}$	114
361	119	112	$\alpha^{21} \leftrightarrow \alpha^{200}$	114
369	47	112	$\alpha^{20} \leftrightarrow \alpha^{228}$	114
391	61	112	$\alpha^{109} \leftrightarrow \alpha^{241}$	114
397	5	112	$\alpha^{17} \leftrightarrow \alpha^{143}$	114
425	185	112	$\alpha^{105} \leftrightarrow \alpha^{181}$	114
451	220	112	$\alpha^{32} \leftrightarrow \alpha^{160}$	114
463	97	112	$\alpha^{54} \leftrightarrow \alpha^{253}$	114
487	187	112	$\alpha^{65} \leftrightarrow \alpha^{198}$	114
501	10	112	$\alpha^{46} \leftrightarrow \alpha^{137}$	114

Table 3.2: Comparison of non-linearities for $n = 8$ by Algorithm 2

Defining Polynomial (Integer value)	**Primitive element (Integer value)**	**Non-linearity of function in [11]**	**Iterations (root↔support)**	**Non-linearity of improved function**
301	2	108	1st $\alpha^{54} \leftrightarrow \alpha^{226}$ 2nd $\alpha^{108} \leftrightarrow \alpha^{245}$ 3rd $\alpha^{75} \leftrightarrow \alpha^{251}$ 4th $\alpha^{33} \leftrightarrow \alpha^{202}$	112
357	2	112	1st $\alpha^{17} \leftrightarrow \alpha^{253}$ 2nd $\alpha^{16} \leftrightarrow \alpha^{165}$	114
425	2	112	1st $\alpha^{81} \leftrightarrow \alpha^{241}$ 2nd $\alpha^{82} \leftrightarrow \alpha^{210}$ 3rd $\alpha^{83} \leftrightarrow \alpha^{250}$	114

Table 3.3: Different improved functions from same parent function

Defining Polynomial (Integer value)	**Primitive element (Integer value)**	**Non-linearity of function in [11]**	**Different options for swapping elements**	**Non-linearity of improved function**
285	2	112	(i) $\alpha^{104} \leftrightarrow \alpha^{230}$	114
			(ii) $\alpha^{36} \leftrightarrow \alpha^{247}$	114
			(iii) $\alpha^{106} \leftrightarrow \alpha^{153}$	114
351	4	112	(i) $\alpha^{32} \leftrightarrow \alpha^{160}$	114
			(ii) $\alpha^{107} \leftrightarrow \alpha^{238}$	114
			(iii) $\alpha^{54} \leftrightarrow \alpha^{234}$	114
369	47	112	(i) $\alpha^{20} \leftrightarrow \alpha^{228}$	114
			(ii) $\alpha^{5} \leftrightarrow \alpha^{220}$	114
			(iii) $\alpha^{31} \leftrightarrow \alpha^{228}$	114
451	220	112	(i) $\alpha^{20} \leftrightarrow \alpha^{228}$	114
			(ii) $\alpha^{101} \leftrightarrow \alpha^{233}$	114
			(iii) $\alpha^{126} \leftrightarrow \alpha^{235}$	114
501	10	112	(i) $\alpha^{46} \leftrightarrow \alpha^{137}$	114
			(ii) $\alpha^{51} \leftrightarrow \alpha^{254}$	114
			(iii) $\alpha^{58} \leftrightarrow \alpha^{136}$	114

3.4 Summarized Results

The devised algorithms were implemented in MAGMA computational algebra system for $8 \leq n \leq 11$ and all the functions belonging to Carlet-Feng infinite class of Boolean functions [11] were improved using these. Majority of functions were improved using Algorithm 1 by a single pair swap (an element from the support set with a root). The functions that could not be improved by Algorithm 1 were improved by Algorithm 2 within at the most four pair swaps. Table 3.4 demonstrates a comparison of non-linearities of the improved functions with their parent functions in [11] for $8 \leq n \leq 11$.

Table 3.4: Comparison of properties of functions

n	**Degree** $f_{Carlet-Feng}$	**AI** $f_{Carlet-Feng}$	**Non-linearity** $f_{Carlet-Feng}$	**Degree** $f_{Improved}$	**AI** $f_{Improved}$	**Non-linearity** $f_{Improved}$
8	7	4	112	7	4	114
9	8	5	232	8	5	234
10	9	5	478	9	5	480
11	10	6	980	10	6	982

Note: The values of $f_{Carlet-Feng}$ have been taken from [11]. The non-linearity values are average, $nl(f_{Improved}) = (nl(f_{Carlet-Feng}) + 2)$ in all cases.

It was mentioned in [11], the product of the constructed functions with any linear functions $(f * l)$ reduces the degree of the resultant functions to at most $n-2$ in case of even number of variables "n" and at most $n-1$ in case of odd "n". Hence the functions do not fall in the worst case or next to worst case resistance to algebraic and fast

algebraic attacks [24]. Similar analysis was performed on the functions improved by Algorithm 1 and 2 for $8 \leq n \leq 10$ and it was ascertained that the improved functions offer identical behavior to the parent functions when the product with the set of all linear functions was obtained. Results of the analysis are presented in Table 3.5.

Table 3.5: Comparison of degrees of functions $f * l$

n	**Degree** $f_{Improved}$	**Degree** $f_{Improved} * l$
8	7	≥6
9	8	≥8
10	9	≥8

3.5 Conclusion

An effective and efficient method of improving non-linearity of Carlet-Feng infinite class of Boolean functions has been developed. The two algorithms devised have been derived from genetic hill climbing algorithm. Not only do these increase the non-linearity of parent functions but also preserve other cryptographic properties including maximal algebraic degree for balanced functions, optimal algebraic immunity and good resistance to algebraic and fast algebraic attacks as shown in practical results presented. Moreover, the proposed method also increases total number of functions that can be obtained for each number of variables, whilst a particular defining polynomial and primitive element is fixed. Both algorithms have been implemented practically using MAGMA and results presented verify the efficacy of proposed method.

CHAPTER 4

TWO HYBRID CLASSES OF BOOLEAN FUNCTIONS

4.1 Tu-Deng Functions

Z. Tu and Y. Deng used ideas from [11], [17] and [19] to construct an infinite class of balanced Boolean functions (Construction 2 in [40]) that achieved maximal algebraic degree, optimal algebraic immunity and a higher non-linearity than the functions belonging to [11] and [45]. Their construction is explained in the subsequent paragraphs.

Let $n = 2k$, $k \geq 1$ and α be a primitive element of $\mathbb{F}_2^k$. The Boolean function $g : \mathbb{F}_2^k \rightarrow \mathbb{F}_2$ is defined as

$$supp(f) = \left\{1, \alpha, \alpha^2, ..., \alpha^{2^{(k-1)}-1}\right\}. \tag{4.1}$$

Define the function *f(x,y)* on $\mathbb{F}_2^k \times \mathbb{F}_2^k$ as follows

$$f = \begin{cases} g(xy^{2^k-2}) & ; \ if x.y \neq 0 \\ 1 & ; \ if x = 0 \, and \, y \, \epsilon \, \triangle' \\ 0 & ; \ otherwise \end{cases} \tag{4.2}$$

where $\triangle' = \left\{\alpha^i \,|\, i = 2^{(k-1)} - 1, 2^{(k-1)}, ..., 2^k - 2\right\}$. The function $f(x,y)$ is balanced, has maximum possible algebraic degree i.e. $2k-1 = n-1$, has optimal algebraic immunity $= \lceil k \rceil = \lceil \frac{n}{2} \rceil$ and the non-linearity is lower bounded by

$$nl(f) \geq 2^{(2k-1)} - 2^{(k-1)} - 2^{\frac{k}{2}} k \ln 2 - 1. \tag{4.3}$$

The function $g(x,y)$ clearly belongs to Dillon's PS_{ap} class of bent functions presented in [17], while the construction principle for the balanced function *f(x,y)* has been adopted from Dobbertin's idea in [19] with slight modification. The functions belonging to this class were practically verified to have achieved non-linearity very close to the bent functions for number of variables $4 \leq n \leq 18$ (using MAGMA). In [12], however, it was pointed out by C. Carlet that for every linear function $l \, \epsilon \, \mathbb{F}_2^n$, the product $l * f$ was equal to $l * g(xy^{(2^k-2)})$. Since the degree of the bent function $g(xy^{(2^k-2)})$ is $k = \frac{n}{2}$, the

degree of $l*f$ was reduced to at the most $k+1 = \frac{n}{2}+1$, which is actually "next to weakest" resistance against fast algebraic attacks [12], as explained in Sub-section 2.11. Repair of these functions was also suggested in [12] by removing the affine components using affine hyper planes. The lower bound for non-linearity of the functions was revised to

$$nl(f') = 2^{(2k-1)} - 2^k = 2^{(n-1)} - 2^{\frac{n}{2}} \tag{4.4}$$

where f' is the repaired function. However, it was commented that the properties of the repaired function including algebraic degree and resistance to fast algebraic attacks were under investigation.

4.2 The two hybrid classes

Before we present the classes constructed, we review the basic ideas that have been utilized for constructing the hybrid classes of functions. We state Dillon's Construction [17] of PS_{ap} class of bent functions based on difference sets, followed by Dobbertin's Construction [19] for obtaining a balanced and highly non-linear Boolean function using a normal bent function.

Theorem 4.1 (Dillon's PS_{ap} class of bent functions [17])**.** *The non-zero points lying on any $2k-1$ lines through the origin constitute a difference set in the affine plane $L \oplus L, L = GF(2^k)$. The bent functions (i.e. characteristic functions) corresponding to these difference sets are equivalent to functions of the form*

$$f(x,y) = Tr(\pi(XY^{2^k-2})) \tag{4.5}$$

where $Tr\{.\}$ is the trace with respect to L/F and $n : L \to L$ is any function for which $Tr(\pi(Z))$ is a balanced function on L which vanishes at 0 (in particular, π may be taken to be any permutation fixing 0). The algebraic degree of this class of bent functions is $deg(f(x,y)) = k$ and hamming weight is $wt(f(x,y)) = 2^{(2k-1)} - 2^{(k-1)}$.

Definition 4.1. A Bent function on $\mathbb{F}_2^{2k}$ is called "Normal" if it is constant on a subspace of $\mathbb{F}_2^{2k}$ of dimension k.

Theorem 4.2 (Dobbertin's balanced and highly non-linear function [19])**.** *Let $W = GF(2^n)$ and $V = W^2$. Let g be a normal bent function on V. That is w.l.o.g. $g(x,0) = 0 \,\forall\, x \in W$. Furthermore, let a balanced function $h : W \to GF(2)$ be given. Set for $x, y \in W$*

$$f(x,y) = \begin{cases} g(x,y) & ; \quad if\ y \neq 0 \\ h(x) & ; \quad otherwise \end{cases} \tag{4.6}$$

then f(x,y) is balanced and we have

$$W_{f_{(a,b)}} = \begin{cases} W_{g_{(a,b)}} + W_{h_{(a,b)}} & ; \quad if\ a \neq 0 \\ 0 & ; \quad otherwise \end{cases} . \tag{4.7}$$

In particular, it follows that

$$R_\Theta = 2^n + R_\theta \tag{4.8}$$

where R is the Spectral Radius of a Boolean function $f : GF(2^n) \rightarrow GF(2)$ *defined as*

$$R_f = \max\left\{|W_f(\alpha)| : \alpha \epsilon \mathbb{F}_2^n\right\} \tag{4.9}$$

Theorem 4.3 ([37])**.** *Let* $h \epsilon \mathfrak{B}_k$ *has* $deg(h) = d$, $1 \leq d \leq k$. *Let g(x,y) be a 2k variable normal bent function with* $deg(g) < k + d$. *Then the 2k variable Boolean Function defined by*

$$f(x,y) = \begin{cases} g(x,y) & ; \quad if\ x \neq 0 \\ h(y) & ; \quad if\ x = 0 \end{cases} \tag{4.10}$$

with $(x,y) \epsilon \mathbb{F}_2^k$ *has* $deg(f(x,y)) = k + d$.

Now we present the two classes constructed using ideas in [11], [40][17] and [19], so that the functions attain balanced-ness, maximal algebraic degree, optimal algebraic immunity and very high non-linearity and do not possess the weakness as functions in [40] that was described in [12]. Our first construction is obtained by modification of Construction 2 in [40], that is itself based on the main idea presented in [19]. We construct a new class of functions in a manner that preserves the other cryptographic properties of Construction 2 in [19], in addition to eliminating the weakness against algebraic and fast algebraic attacks. The class also includes a considerably larger total number of functions for each number of variables using function belonging to the infinite class in [11] as the balanced component. Our second construction is a novel construction as it differs from Dobbertin's main idea in [19] and Construction 2 of [40] significantly. This construction includes even a larger total number of functions in comparison with our first construction and also uses a function different than the ones in [11] as the balanced component.

Theorem 4.4 (**Construction 1**)**.** *Let g and h be two balanced Boolean functions* $g, h : \mathbb{F}_2^k \rightarrow \mathbb{F}_2$ *defined as*

$$supp(g) = \left\{1, \alpha, \alpha^2, ..., \alpha^{2^{(k-1)}-1}\right\} \tag{4.11}$$

$$supp(h) = \left\{\alpha^j, \alpha^{j+1}, ..., \alpha^{2^{(k-1)}+j-1}\right\}\ ;\ for\ 1 \leq j \leq 2^{k-2} - 1 \tag{4.12}$$

where α *is a primitive element of* $\mathbb{F}_2^k$ *, g has the same definition as in [40], while h is a function belonging to the infinite class in [11], where it is clearly mentioned that support set of functions introduced in that class can, in fact, be defined for every n as* $\left\{\alpha^j, \alpha^{j+1}, ..., \alpha^{2^{(n-1)}+j-1}\right\}$ *for a suitable j, while maintaining optimal algebraic immunity.*

Note that for $j = 0$, the function $h = g$, while for $j = 2^{(k-1)} - 1$, h equals the conditions used to balance construction in [40] and we avoid using this value. Now for $n = 2k \geq 4$, define the function $f_1(x, y) : \mathbb{F}_2^k \times \mathbb{F}_2^k \to \mathbb{F}_2$ as

$$f_1(x,y) = \begin{cases} g(xy^{-1}) & ; \quad if\ x \neq 0\ and\ y \neq 0 \\ h(x) & ; \quad otherwise \end{cases} \tag{4.13}$$

then $f_1(x, y)$ is a balanced Boolean function with maximum possible algebraic degree $= 2k - 1 = n - 1$, optimal algebraic immunity $= \lceil k \rceil = \lceil \frac{n}{2} \rceil$, and very high non-linearity.

The comparison of non-linearity with [11] and [40] is presented in Table 4.2. It has been practically verified for $4 \leq n \leq 12$ (using MAGMA) that this construction does not have the weakness as the function in [40] pointed out in [12], details are discussed in Section 4.4. A comparison with functions in [40] has been presented in Table 4.4.

Theorem 4.5 (**Construction 2**). *Let g and h be two balanced Boolean functions $g, h : \mathbb{F}_2^k \to \mathbb{F}_2$ defined as*

$$supp(g) = \left\{1, \alpha, \alpha^2, ..., \alpha^{2^{(k-1)}-1}\right\} \tag{4.14}$$

$$supp(h) = \left\{0, \alpha^j, \alpha^{j+1}, ..., \alpha^{2^{(k-1)}+j-2}\right\} \ ; \ for\ 1 \leq j \leq 2^{k-1} \tag{4.15}$$

where α is a primitive element of $\mathbb{F}_2^k$, Again, g has the same definition as in [40], however, h does not belong to the infinite class in [11]. Now for $n = 2k \geq 4$, define the function $f_2(x, y) : \mathbb{F}_2^k \times \mathbb{F}_2^k \to \mathbb{F}_2$ as

$$f_2(x,y) = \begin{cases} g(xy^{-1}) & ; \quad if (x \neq 0\ and\ y \neq 0)\ and\ (x \neq y) \\ 0 & ; \quad if\ x \neq 0\ and\ y = 0 \\ h(x) & ; \quad otherwise \end{cases} \tag{4.16}$$

then $f_2(x, y)$ is a balanced Boolean function with maximum possible algebraic degree $= 2k - 1 = n - 1$, optimal algebraic immunity $= \lceil k \rceil = \lceil \frac{n}{2} \rceil$, and very high non-linearity.

Our Construction 1 is a modification of Construction 2 of [40] using the function h belonging to the class in [11] as the balanced function but takes more output values from the balanced function than the normal bent one as compared to [40]. Our Construction 2 is a novel construction that is different to even the main idea presented by Dobbertin in [19]. $f_2(x, y)$ in our Construction 2 differs from the normal bent function $g(xy^{-1})$ on more input vectors than the function proposed in [19], [40] and even our Construction 1. Moreover, the function h is also different to the infinite class in [11] and has much more flexibility in construction as compared to our Construction 1, due to larger range for j. These functions also do not possess the weakness pointed out in [12] and have been practically verified for $4 \leq n \leq 12$ (using MAGMA). Table 4.2 highlights the comparison with [11] and [40] in terms of non-linearity of the functions, while Table 4.4 indicates the comparison with respect to the weakness [12].

4.3 Analysis of the Constructions 1 and 2

We shall now analyse the balanced-ness, non-linearity, algebraic degree, algebraic immunity and the resistance to algebraic and fast algebraic attacks of the functions in Constructions 1 and 2.

Proposition 4.6. *The functions belonging to Constructions 1 and 2 are balanced.*

Proof. Using the result in 4.1, the hamming weight of the function $g(xy^{-1})$ is $2^{(2k-1)} - 2^{(k-1)}$. Furthermore, $h(x)$ is a balanced Boolean function on $\mathbb{F}_2^k$ and therefore, has weight $2^{(k-1)}$. Hence the weight of functions f_1 and f_2 is $wt(f_1, f_2) = wt(g(xy^{-1})) + wt(h(x)) = 2^{(2k-1)} - 2^{(k-1)} + 2^{(k-1)} = 2^{(2k-1)} = 2^{(n-1)}$. Therefore, the functions are balanced. □

Proposition 4.7. *The non-linearity of function in Construction 1 and 2 satisfies*

$$nl(f) \geq 2^{(2k-1)} - 2^{(k-1)} - 2^{\frac{k}{2}} k \ln 2 - 1. \tag{4.17}$$

Proof. It is obvious that the function $g(xy^{-1})$ is a normal bent function as per definition. Hence we first use 4.2 to show that $nl(f_1, f_2) = 2^{(n-1)} - 2^{\frac{n}{2}} + nl(h)$. In this theorem, it was established that

$$W_{f_{(a,b)}} = \begin{cases} W_{g_{(a,b)}} + W_{h_{(a,b)}} & ; \quad if\ a \neq 0 \\ 0 & ; \quad otherwise \end{cases}. \tag{4.18}$$

Moreover, it has been already established that the dual of any normal bent function is also normal (Lemma 7 of [19]). Using the fact that a normal bent function like $g(xy^{-1})$ on $\mathbb{F}_2^n$ that is constant on an affine sub-space S of $\mathbb{F}_2^n$ with dimension $S = \frac{n}{2}$, is also constant on each proper coset of S [19], we deduce that the function has $2^{\frac{n}{2}-1}$ values of $2^{\frac{n}{2}}$ and $2^{\frac{n}{2}-1}$ values of $-2^{\frac{n}{2}}$ in the Walsh spectrum. Hence for a fixed x_0, $W_g(x_0, y)$ is $\pm 2^{\frac{n}{2}}$ and the non-linearity of f can be computed as

$$\begin{aligned} nl(f_1, f_2) &= 2^{(n-1)} - \frac{1}{2}\left(2^{\frac{n}{2}} + \max_{\alpha \epsilon \mathbb{F}_2^k} |W_h(\alpha)|\right) \\ &= 2^{(n-1)} - \frac{1}{2}\left(2^{\frac{n}{2}} + 2^{\frac{n}{2}} - 2.nl(h)\right) \\ &= 2^{(n-1)} - 2^{\frac{n}{2}} + nl(h). \end{aligned} \tag{4.19}$$

In [40], the lower bound on non-linearity of their construction has been computed as

$$nl(f) \geq 2^{(2k-1)} - 2^{(k-1)} - 2^{\frac{k}{2}} k \ln 2 - 1.$$

Since the function $g(xy^{-1})$ is the same as in Construction 2 of [40] and the support set of $h(x)$ also has the same dimension, the proof for non-linearity of functions in

Proposition 5.4 of [40] remains valid and we do not repeat it here. It may be noted that the above inequality gives a lower bound on the non-linearities of the functions in Constructions 1 and 2, but the exact non-linearities can be precisely calculated using Equation 4.19 once that of the function $h(x)$ is known. □

Proposition 4.8. *The algebraic degree of functions in Constructions 1 and 2 is*

$$deg(f_1, f_2) = 2k - 1 = n - 1. \tag{4.20}$$

Proof. Since we have ascertained that $g(xy^{-1})$ is a normal bent function, using the result of 4.3, we get

$$deg(f_1, f_2) = deg(g(xy^{-1})) + deg(h(x)). \tag{4.21}$$

From Remark 6.3.11 of [17] and using the fact that PS_{ap} bent functions are a sub-class of $PS^{(-)}$ class, we have $deg(g(xy^{-1}))$ defined over $\mathbb{F}_2^{2k} \approx \mathbb{F}_2^n$ as $deg(g(xy^{-1})) = k = \frac{n}{2}$. In [11], the degree of $h(x)$ defined over $\mathbb{F}_2^k$ has been proved to be $deg(h(x)) = k - 1$. Hence, we have

$$deg(f_1, f_2) = k + k - 1 = 2k - 1 = n - 1.$$

□

Proposition 4.9. *With the assumption that Tu-Deng conjecture in [40] is correct; the algebraic immunity of functions in Construction 1 and 2 is optimal i.e.* $\lceil \frac{n}{2} \rceil$.

Proof. Since there is no change in the support of the function $g(xy^{-1})$ defined in Construction 1 of [40] and the support set of $h(x)$ also has the same dimension, the proof of algebraic immunity in Proposition 5.1 of [40] also remains valid, so we do not reproduce it here. □

4.4 Implementation and results

As mentioned earlier, the functions in two proposed hybrid classes attain very high non-linearity values. In fact, they maintain the close to bent function non-linearities as in [40]. Owing to the structure of the balanced function $h(x)$ used in our constructions (the range of j in its support set), the total number of functions for each number of variables n increases quite significantly. Most importantly, it has been practically verified for all linear functions $l \epsilon \mathfrak{B}_n$ that the degree of $l * f_1$ and $l * f_2$ is at least $2k - 2 = n - 2$ for even values of k and $2k - 1 = n - 1$ for odd values of k for the range $4 \leq n \leq 12$, and we conjecture it for all even n. Hence there exist no non-zero function g of degree $\leq e$ and no function h of degree at the most d such that $f * g = h$, when $(e, d) = (1, n - 2)$ for odd k and $(1, n - 3)$ when k is even.

All implementations were done in MAGMA, including construction of the two hybrid classes in Construction 1 and 2, computation of non-linearity and algebraic degree for $4 \leq n \leq 18$, and analysis of resistance to weakness pointed out in [12] for $4 \leq n \leq 12$. The "boolfun" library of R-package was utilized to verify the results for algebraic immunity of constructed functions for $4 \leq n \leq 12$. Table 4.1 highlights a comparison between values of non-linearity of functions belonging to Construction 1 and 2 (in MAGMA) with the lower bound as per Proposition 4.7. Table 4.2 gives the comparison in terms of non-linearities of the functions in [11] and [40] with our hybrid classes. A count of total number of functions possible in [40] and our constructions is depicted in Table 4.3. Finally, Table 4.4 demonstrates the results of product of the functions belonging to our hybrid classes with the set of all linear functions $l \epsilon \mathfrak{B}_n$ and compares it with the functions in [40].

Table 4.1: Comparison of non-linearities in Prop 4.7 and constructed functions

n	**nl**(f_1, f_2) **in Proposition 5.2**	**nl**(f_1, f_2) **constructed**
4	≥3	4
6	≥21	26
8	≥107	116
10	≥476	490
12	≥1982	2008
14	≥8073	8118
16	≥32551	32624
18	≥130674	130792

Table 4.2: Comparison of non-linearities with [11] and [40]

n	**nl**(f_{bent})	**nl**$(f_{[\mathbf{11}]})$	**nl**$(f_{[\mathbf{40}]})$	**nl**(f_1, f_2)
4	6	4*	4	4
6	28	24	26	26
8	120	112	116	116
10	496	478	490	490
12	2016	1970*	2008	2008
14	8128	8036*	8118	8118
16	32640	32530*	32624	32624
18	130812	130442*	130792	130792

The values with * have been computed in [40] and not in the original construction in [11]

Table 4.3: Comparison of total number of functions with [40]

n	**No of primitive elements(α)**	**No of functions in [40]**	**No of functions in Construction 1 and 2 (1/2)**
4	2	2	2/2
6	6	12	12/48
8	8	24	72/192
10	30	180	1260/2880
12	36	324	4860/10368
14	126	2268	70308/145152
16	128	3840	241920/491520
18	432	23760	3017520/6082560

Table 4.4: The comparison of degrees of $f * l$ with [40]

n	**deg(f_{TuDeng})**	**deg($l * f_{TuDeng}$)***	**deg(f_1, f_2)**	**deg($l * f_1, f_2$)**
4	3	≤ 2	3	≥ 2
6	5	≤ 4	5	≥ 5
8	7	≤ 5	7	≥ 6
10	9	≤ 6	9	≥ 9
12	11	≤ 7	11	≥ 10

The column with $\star$ indicates values have been calculated as pointed out in [12]

4.5 Conclusion

We have presented two hybrid classes of Boolean functions for an even $n \geq 4$, derived from ideas in [11],[40], [17] and [19] to construct balanced Boolean functions with maximum possible algebraic degree, optimal algebraic immunity and very high values of non-linearity. While our Construction 1 is a modification of Construction 2 in [40], our Construction 2 is a new one that significantly differs from classes proposed in [11], [40] and [19]. Furthermore, the functions belonging to these classes are not weak against fast algebraic attacks as were the functions in [40] indicated in [12]. We have practically implemented and verified the described cryptographic properties of these classes for $4 \leq n \leq 18$, while the removal of the weakness pointed out in [12] has been confirmed for $4 \leq n \leq 12$. The implementations were done using MAGMA, while the verification of properties and calculation of algebraic immunity was performed using "boolfun" library of R-package.

CHAPTER 5

A CLASS OF 1-RESILIENT FUNCTIONS

5.1 Constructing Resilient functions

The infinite classes of functions proposed in [11], [19] and our Construction 1 and 2 are all 0-resilient and are suited for application in block ciphers. In case of stream cipher on the other hand, constructions like non-linear combining, non-linear filtering and alternating step generator require functions to be at least 1-resilient. However, increase in resilience results in reduction in at least one property out of non-linearity, algebraic degree and algebraic immunity of functions.

Siegenthaler's inequality in [46] establishes that the upper bound on algebraic degree of a m-resilient function of "n" variables is "$n - m - 1$". It means that for a 1-resilient function, maximum possible algebraic degree is "$n - 2$", that is one less than maximal degree achieved in our constructions 1 and 2. It also implicates that a maximum achievable order of resilience is $m = n - 2$, which means that the algebraic degree of this function would be 1, that is it would be a linear function. Additionally, since Parseval's Indentity (2.5) demands conservation of energy, increase in number zero entries in Walsh spectrum to improving resilience implies that the maximum absolute value may increase. This decreases the non-linearity of function (2.6). The upper bounds for non-linearity of m-resilient functions ([34, 36, 41]) are

$$nl(f) \leq \begin{cases} 2^{n-1} - 2^{m+1} & ; \quad if\ n/2 - 1 < m + 1 \\ 2^{n-1} - 2^{n/2-1} - 2^{m+1} & ; \quad if\ n = even\ \&\ n/2 - 1 \geq m + 1 \\ 2^{n-1} - 2^{m+1}\lceil 2^{n/2} - m - 2 \rceil & ; \quad if\ n = odd\ \&\ n/2 - 1 \geq m + 1 \end{cases} .$$

The trade-off between resilience and achievable non-linearity is tabulated in [36], which highlights the upper bound for non-linearity of m-resilient functions for $m \geq 1$ that clearly demonstrate a reduction in achievable non-linearity as compared to 0-resilient ones. Nevertheless, we propose a class of 1-resilient functions in n-variables over $\mathbb{F}_2^n$ by using ideas in [37, 39] and modifying our Construction 2.

Theorem 5.1 (**Construction 3**). *Let g and h be two balanced Boolean functions* $g, h : \mathbb{F}_2^k \to \mathbb{F}_2$ *defined as*

$$supp(g) = \left\{1, \alpha, \alpha^2, ..., \alpha^{2^{(k-1)}-1}\right\} \tag{5.1}$$

$$supp(h) = \left\{0, \alpha^j, \alpha^{j+1}, ..., \alpha^{2^{(k-1)}+j-2}\right\} \; ; \; for \; 1 \leq j \leq 2^{k-1} \tag{5.2}$$

where α *is a primitive element of* $\mathbb{F}_2^k$. *The function g has the same definition as in [40] but the function h does not belong to the infinite class in [11]. Now for* $n = 2k \geq 4$, *define the function* $f_3(x, y) : \mathbb{F}_2^k \times \mathbb{F}_2^k \to \mathbb{F}_2$ *as*

$$f_3(x,y) = \begin{cases} g(xy^{-1}) & ; \; if \; (x \neq 0 \, and \, y \neq 0) \, and \, (x \neq y) \\ h(x) & ; \; if \; x = y \neq 0 \\ h(x) \oplus 1 & ; \; if \; y = 0 \\ h(y) \oplus 1 & ; \; if \; x = 0 \end{cases} \tag{5.3}$$

then $f_3(x, y)$ *is a 1-resilient Boolean function with high algebraic degree* $= 2k - 2 = n - 2$, *optimal algebraic immunity* $= \lceil k \rceil = \lceil \frac{n}{2} \rceil$, *and very high non-linearity.*

5.2 Analysis of Construction 3

The proofs for balanced-ness and algebraic degree remain unchanged since the support set of g, h and construction of the normal bent function $g(xy^{-1})$ remain the same as our Construction 2. We were not able to prove optimal algebraic immunity although we have verified it practically by implementation in MAGMA. We leave the proof of optimal algebraic immunity as an open problem. Hence, we now need to prove only 1-resilience of functions belonging to this construction.

Proposition 5.2. *The functions belonging to Constructions 3 are 1-resilient.*

Proof. We note that for 1-resilience, $W_f(\lambda) = 0 \, \forall \, \lambda$ such that $wt(\lambda) \leq 1$. Since the functions are balanced, $W_f(0) = 0$ and we only need to investigate the cases when $wt(\lambda) = 1$. According to Construction 3, there are only two such cases; $a \neq 0, b = 0$ and $a = 0, b \neq 0$. Hence, we start by the following fact

$$Supp(f_3(a,b)) = \begin{cases} \{\beta b, b\} & ; \; \beta \in Supp(g) \notin \{1\}, \; b \in \mathbb{F}_2^{k^*} \\ \{a, b\} & ; \; a = b \in Supp(h) \notin \{0\} \\ \{a, 0\} & ; \; a \in Supp(h \oplus 1) \\ \{0, b\} & ; \; b \in Supp(h \oplus 1) \end{cases} . \tag{5.4}$$

Now let's calculate the Walsh spectrum of $f_3(a, b)$

$$W_{f_3}(a,b) = \sum_{x,y\epsilon\mathbb{F}_2^k} (-1)^{f_3(x,y)}(-1)^{tr(ax+by)}.$$

We know that $(-1)^{f(x)} = 1 - 2f(x)$, therefore

$$\begin{aligned} W_{f_3}(a,b) &= \sum_{x,y\epsilon\mathbb{F}_2^k} (1 - 2f_3(x,y))(-1)^{tr(ax+by)} \\ &= \sum_{x,y\epsilon\mathbb{F}_2^k} (-1)^{tr(ax+by)} - 2\sum_{x,y\epsilon\mathbb{F}_2^k} f_3(x,y)(-1)^{tr(ax+by)} \\ &= \sum_{x,y\epsilon\mathbb{F}_2^k} (-1)^{tr(ax+by)} - 2\sum_{x,y\epsilon Supp(f_3(x,y))} (-1)^{tr(ax+by)}. \end{aligned}$$

Recall that

$$\sum_{x\epsilon\mathbb{F}_2^k} (-1)^{tr(ax)} = \begin{cases} 2^k & ; \ if\ a = 0 \\ 0 & ; \ otherwise \end{cases}. \tag{5.5}$$

Therefore

$$\begin{aligned} W_{f_3}(a,b) &= -2 \sum_{x,y\epsilon Supp(f_3(x,y))} (-1)^{tr(ax+by)} \\ &= -2 \sum_{\beta\epsilon Supp(g)\not\in\{1\}} \sum_{y\epsilon\mathbb{F}_2^{k^*}} (-1)^{tr(a\beta+b)y} - 2 \sum_{y\epsilon Supp(h(y))\not\in\{0\}} (-1)^{tr(a+b)y} \\ &\quad - 2 \sum_{x\epsilon Supp(h(x)\oplus 1)} (-1)^{tr(ax)} - 2 \sum_{y\epsilon Supp(h(y)\oplus 1)} (-1)^{tr(by)} \\ &= -2 \sum_{\beta\epsilon Supp(g)\not\in\{1\}} \left[\sum_{y\epsilon\mathbb{F}_2^k} (-1)^{tr(a\beta+b)y} - 1 \right] - 2 \sum_{y\epsilon Supp(h(y))\not\in\{0\}} (-1)^{tr(a+b)y} \\ &\quad - 2 \sum_{x\epsilon Supp(h(x)\oplus 1)} (-1)^{tr(ax)} - 2 \sum_{y\epsilon Supp(h(y)\oplus 1)} (-1)^{tr(by)}. \end{aligned} \tag{5.6}$$

Now using $a \neq 0, b = 0$ in 5.6, we get

$$
\begin{aligned}
W_{f_3}(a,b) &= -2\sum_{\beta \epsilon Supp(g)\notin\{1\}}\left[\sum_{y\epsilon\mathbb{F}_2^k}(-1)^{tr(a\beta y)}-1\right] - 2\sum_{y\epsilon Supp(h(y))\notin\{0\}}(-1)^{tr(ay)} \\
&\quad - 2\sum_{x\epsilon Supp(h(x)\oplus 1)}(-1)^{tr(ax)} - 2\sum_{y\epsilon Supp(h(y)\oplus 1)}(-1)^0 \\
&= -2\sum_{\beta \epsilon Supp(g)\notin\{1\}}\left[\sum_{y\epsilon\mathbb{F}_2^k}(-1)^{tr(a\beta y)}-1\right] - 2\left[\sum_{x\epsilon\mathbb{F}_2^k}(-1)^{tr(ax)} - 1\right] \\
&\quad - 2\sum_{y\epsilon Supp(h(y)\oplus 1)}(1).
\end{aligned}
$$

Since $Supp(h(x)) = Supp(h(x)\oplus 1) = 2^{k-1}$, using 5.5 we get

$$
\begin{aligned}
W_{f_3}(a,b) &= -2\sum_{\beta \epsilon Supp(g)\notin\{1\}}[0-1] - 2[0-1] - 2.2^{k-1} \\
&= 2(2^{k-1}-1) + 2 - 2.2^{k-1} \\
&= 0.
\end{aligned} \tag{5.7}
$$

The case for $a = 0, b \neq 0$ is exactly the same as above with change of exponents only. Hence it is proved that functions belonging to construction 3 are 1-resilient.

□

5.3 Implementation and results

Construction 3 was also implemented in MAGMA and results were obtained for $4 \leq n \leq 18$. The results were then compared with already proposed constructions that achieved optimum values of algebraic degree, algebraic immunity and high non-linearity of functions. In all cases, maximum possible algebraic degree for a 1-resilient Boolean function i.e. $n-2$ is achieved. A comparison of rest of the properties with constructions in [37, 39] is presented in Table 5.1.

Table 5.1: Comparison of our Construction 3 with [37, 39]

n	**nl**($f_{[\mathbf{39}]}$)	**AI**($f_{[\mathbf{39}]}$)	**nl**($f_{[\mathbf{37}]}$) **Thm 9/10**	**AI**($f_{[\mathbf{37}]}$)	**nl**(f_3)	**AI**(f_3)
4	4	2	4	1*	4	2
6	24	3	22/18	2*	24	3
8	112	4	108/103	3*	112	4
10	484	5	484/482	4*	484	5
12	1996	6	1998/1994	5*	1996	6
14	8100	7	8104/8106	6*	8100	7
16	32588	8	32604/-	7*	32588	8
18	130760	9	130768/130778	8*	130760	9

The entries with $*$ indicate that the functions have at least sub-optimal values of algebraic immunity, as indicated by authors themselves in [37]

It is evident from above comparison that our Construction 3 achieves the best results for 1-resilient functions having optimal algebraic immunity,together with [39]. The construction in [37] achieves better non-linearities for $n \geq 12$ but they do not guarantee optimal algebraic immunity, whereas the construction in [39] and our Construction 3 achieve optimal algebraic immunity (practically verified in MAGMA). The total number of functions in our class is also much larger as compared to [39]. A comparison between total number of functions for each n is presented in Table 5.2. Moreover, constructions in [37, 39] do not investigate the resistance of functions to algebraic and fast algebraic attacks. We performed similar analysis of our Construction 3 as for Construction 1 and 2 to test resistance to these attacks; results are depicted in Table 5.3 which clearly reflect that functions in this class offer good resistance to algebraic and fast algebraic attacks.

Table 5.2: Total number of functions in [39] and our construction 3

n	**No of primitive elements**(α)	**No of functions in [39]**	**No of functions in Construction 3**
4	2	2	2
6	6	12	48
8	8	24	192
10	30	180	2880
12	36	324	10368
14	126	2268	145152
16	128	3840	491520
18	432	23760	6082560

Table 5.3: Degrees of $f * l$ for our Construction 3

n	**deg**(f_3)	**deg**($l * f_3$)
4	2	2
6	4	5
8	6	7
10	8	9
12	10	11

5.4 Obtaining m-Resilient functions

As discussed earlier, increasing resilience of functions requires a trade-off with achievable algebraic degree, non-linearity and algebraic immunity. Having said that, certain applications, such as functions used as non-linear combining or non-linear filtering functions, do require a reasonable order of resilience in order to effectively resist correlation and fast correlation attacks. Given a t-resilient functions, many methods of obtaining m-resilient functions for $m \geq (t + 1)$ have been proposed based on composition of functions, iterative and recursive approaches. For example, constructions in [29, 38, 44] result in increase in order of resilience and number of variables, those in [10, 31] obtain functions with same order of resilience but larger number of variables, while methods in [20, 35] can be used to construct different functions with same order of resilience and number of variables from known ones.

This implies that using our construction 3 as the base class, functions with higher order of resilience can be easily constructed using methods proposed in [29, 38, 44]. Another interesting implication is that although our hybrid class in construction 3 is for even "n" only, techniques in [7, 10, 29, 31, 32, 38, 44] can be used to increase the order of resilience as well as obtain functions for odd "$n + 1$" variables as well. For instance, construction in [38] can be used to obtain m-resilient functions in "$n + 1$" variables and $(m + 1)$-resilient functions in "$n + 2$" variables using two m-resilient functions in "n" variables as base functions. Similarly, two functions on "n_1" and "n_2" variables that are m_1 and m_2-resilient respectively, can be used as base functions to construct $(m_1 + m_2 + 1)$-resilient function in "$n_1 + n_2$" variables using construction in [44].

5.5 Conclusion

We have proposed a class of 1-resilient functions in Construction 3 with high algebraic degree, optimal algebraic immunity and high non-linearity by using ideas in [37, 39] and modifying our hybrid class Construction 2 (Theorem 4.5). Functions in our construction 3 can be utilized as base functions to obtain m-resilient functions for $m \geq 2$ in number of variables $\geq n$ using methods proposed in [7, 10, 29, 31, 32, 38, 44]. We have practically implemented and verified the described cryptographic properties of this class for $4 \leq n \leq 18$. We have also tested the functions belonging to this class for

resistance against algebraic and fast algebraic attacks for $4 \leq n \leq 12$. Results show that Construction 3 also offers good resistance against these attacks. The implementations were done using MAGMA, while the verification of properties and calculation of algebraic immunity was performed using "boolfun" library of R-package.

CHAPTER 6

CONCLUSION

In this thesis, we have proposed two algorithms by modifying the genetic hill climbing algorithm [40] for improvement in the non-linearity of functions constructed using the infinite class proposed in [11] for number of variables $n \geq 8$. The improved functions achieve higher non-linearity than the original functions in [11] and also maintain high algebraic degree and optimal algebraic immunity. The proposed algorithms have been verified by constructing all Boolean functions for $8 \leq n \leq 11$ and improving their non-linearity.

We have also constructed two hybrid classes of balanced Boolean functions based on ideas in [11, 17, 19] by modifying the construction in [40]. The modified functions maintain their cryptographic properties i.e. balanced-ness, maximal algebraic degree for balanced functions, optimal algebraic immunity and very high non-linearity and avoid the weakness pointed out against algebraic and fast algebraic attacks in [12]. We have also practically analysed and verified (using MAGMA) that functions belonging to the two proposed hybrid classes are not comparably vulnerable to fast algebraic attacks as functions in [40] and offer good resistance.

Finally, we have constructed a 1-resilient class of Boolean functions with high algebraic degree, optimal algebraic immunity and high non-linearity by by using ideas in [37, 39] and modifying our second construction in Chapter 4. This construction was also implemented in MAGMA and analysed to verify their cryptographic properties. An analysis for resistance against algebraic and fast algebraic attacks was also performed which shows that functions belonging to this class offer good resistance to these attacks. Functions belonging to this class can also be utilized as base functions to obtain m-resilient functions for $m \geq 2$ in number of variables $\geq n$ using methods proposed in [7, 10, 29, 31, 32, 38, 44].

Bibliography

[1] F.Armknecht: Improving Fast Algebraic Attacks. In Fast Software Encryption 2004, LNCS 3017, pages 65-82. Springer Verlag, 2004.

[2] F.Armknecht: Algebraic Attacks and Annihilators. In WEWoRC 2005, volume P-74 of LNI, pages 13-21. Gesellschaftfur Informatik, 2005.

[3] F.Armknecht, and G.Ars: Introducing a New Variant of Fast Algebraic Attacks and Minimizing Their Successive Data Complexity. In Progress in Cryptology - Mycrypt 2005, LNCS 3715, pages 16-32. Springer Verlag, 2005.

[4] W.Bosma, J.J.Cannon, C.Fieker, A.Steel (eds.): Handbook of Magma functions, Edition 2.14.17 (2008), 4546 pages.

[5] A.Braeken, B.Preneel: On the algebraic immunity of symmetric Boolean functions. In: Maitra, S., Veni Madhavan, C.E., Venkatesan, R. (eds.) INDOCRYPT 2005. LNCS, vol. 3797, pp. 35-48. Springer, Heidelberg (2005).

[6] A.Canteaut, M.Trabbia: Improved Fast Correlation attacks using parity-check equations of weight 4 and 5. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 573-588. Springer, Heidelberg (2000).

[7] C.Carlet: Designing bent functions and resilient functions from known ones, without extending their number of variables. Proceedings of International Symposium on Information Theory (ISIT 2005), pp. 1096-1100, September, 2005.

[8] C.Carlet: A method of construction of balanced functions with optimum algebraic immunity. IACR Cryptology ePrint Archive 2006: 149 (2006).

[9] C.Carlet, D.K.Dalai, K.C.Gupta, S.Maitra: Algebraic immunity for cryptographically significant Boolean functions: analysis and construction. IEEE Trans. Inform. Th. 52(7), 3105-3121 (2006).

[10] C.Carlet: On bent and highly nonlinear balanced / resilient functions and their algebraic immunities. Proceedings of AAECC 16, LNCS 3857, pp. 1-28, 2006.

[11] C.Carlet, K.Feng: An Infinite Class of Balanced functions with Optimal Algebraic Immunity, Good Immunity to Fast Algebraic Attacks and Good Nonlinearity. ASIACRYPT 2008: 425-440.

[12] C.Carlet: On a weakness of the Tu-Deng function and its repair. Cryptology ePrint Archive, Report 2009/606. http://eprint.iacr.org/2009/606.

[13] N.Courtois, and W.Meier: Algebraic Attacks on Stream Ciphers with Linear Feedback. In Advances in Cryptology - EUROCRYPT 2003, LNCS 2656, pages 345-359. Springer Verlag, 2003.

[14] N.Courtois: Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. In Advances in Cryptology - CRYPTO 2003, LNCS 2729, pages 176-194. Springer Verlag, 2003.

[15] D.K.Dalai, K.C.Gupta, S.Maitra: Cryptographically significant Boolean functions: construction and analysis in terms of algebraic immunity. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 98-111. Springer, Heidelberg.

[16] D.K.Dalai, S.Maitra, S.Sarkar: Basic theory in construction of Boolean functions with maximum possible annihilator immunity. Des. Codes Cryptogr. 40(1), 41-58 (2006).

[17] J.F. Dillon: Elementary Hadamard Difference Sets. PhD thesis, University of Maryland (1974).

[18] C.Ding, G.Xiao, W.Shan(eds.): The Stability Theory of Stream Ciphers. LNCS, vol. 561. Springer, Heidelberg (1991).

[19] H.Dobbertin: Construction of bent functions and balanced boolean functions with high nonlinearity. Workshop on Fast Software Encryption (LNCS), Springer-Verlag, 1995, vol. 1008, pp. 61-74.

[20] J.C.Faugére: A new efficient algorithm for computing Gröbner bases (F4), Journal of Pure and Applied Algebra 139 (1999), 61-88.

[21] J.C.Faugére: A new efficient algorithm for computing Gröbner bases without reduction to zero (F5), International Symposium on Symbolic and Algebraic Computation (ISSAC 2002), ACM Press, 2002, pp. 75-83.

[22] J.C.Faugére and G. Ars: An algebraic cryptanalysis of nonlinear filter generators using Grö bner bases, Rapport de recherche 4739, February 2003, www.inria.fr/rrrt/rr-4739.html.

[23] N.Li, W.Qi: Construction and analysis of boolean functions of 2t+1 variables with maximum algebraic immunity. In: Advances in Cryptology, Asiacrypt 2006. LNCS, vol. 4284, pp. 84-98 (2006).

[24] N.Li, L.Qu, W.Qi, G.Feng, C.Li, D.Xie: On the construction of Boolean functions with optimal algebraic immunity. IEEE Trans. Inform. Theory 54, 1330-1334 (2008).

[25] M.Lobanov: Tight bound between nonlinearity and algebraic immunity. Paper 2005/441 (2005), http://eprint.iacr.org/.

[26] W.Meier, E.Pasalic, C.Carlet: Algebraic attacks and decomposition of Boolean functions. In: Goos G, Hartmanis J, van Leeuwen J, eds. Advances in Cryptology-EUROCRYPT 2004. LNCS, Vol. 3027. Berlin: Springer-Verlag, 2004. 474-491.

[27] W.Meier, O.Staffelbach: Fast correlation attacks on stream ciphers. In: GÂ¨unther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 301-314. Springer, Heidelberg.

[28] S.Pan , X.Fu, W.Zhang: Construction of 1-Resilient Boolean functions with Optimal Algebraic Immunity and Good Nonlinearity. Journal of Computer Science and Technology, Volume 26 Issue 2, March 2011.

[29] E.Pasalic, S.Maitra, T.Johansson, P.Sarkar: New Constructions of Resilient and Correlation Immune Boolean Functions Achieving Upper Bound on Nonlinearity. WCC2001, International Workshop on Coding and Cryptography. Electronic Notes in Discrete Mathematics, Volume 6, pp 158-167, April 2001.

[30] S.Rønjom, T.Helleseth: A new attack on the filter generator. IEEE Trans. on Inform. Th. 53(5), 1752-1758 (2007).

[31] P.Sarkar, S.Maitra: Nonlinearity Bounds and Constructions of Resilient Boolean Functions. Advances in Cryptology - CRYPTO 2000, Lecture Notes in Computer Science, Volume 1880, 2000, pp 515-532.

[32] P.Sarkar, 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 (LNCS), pp. 485-506, 2000.

[33] T.Siegenthaler: Correlation Immunity of Nonlinear Combining Functions for Cryptographic Applications. IEEE Trans. on Inform. Th. IT-30(5), 776-780 (1984).

[34] C.E. Shannon: Communication theory of secrecy systems, Bell System Technical Journal 28 (1949), no. 4, 656-715.

[35] T. Siegenthaler: Decrypting a class of stream ciphers using ciphertext only, IEEE Transactions on Computers C-34 (1985), no. 1, 81-85.

[36] D.Tang, C.Carlet, X.Tang: Highly Nonlinear Boolean Functions with Optimal Algebraic Immunity and Good Behavior Against Fast Algebraic Attacks in IEEE Transactions on Information Theory, Volume:PP, Issue 99.

[37] X.Tang, D.Tang, X.Zeng, L.Hu: Balanced Boolean functions with (Almost) Optimal Algebraic Immunity and Very High Nonlinearity. Cryptology ePrint Archive, Report 2010/443. http://eprint.iacr.org/2010/443.

[38] Y. V. Tarannikov: On resilient Boolean functions with maximum possible nonlinearity. Proceedings of INDOCRYPT 2000, Lecture Notes in Computer Science 1977, pp. 19-30, 2000.

[39] Z.Tu, Y.Deng: A Class of 1-Resilient Function with High Nonlinearity and. Algebraic Immunity. IACR Cryptology ePrint Archive 2010, 179 (2010).

[40] Z.Tu, Y.Deng: A conjecture about binary strings and its applications on constructing Boolean functions with optimal algebraic immunity. Des. Codes Cryptogr., 2010. Online First Articles. DOI 10.1007/s10623-010-9413-9.

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

[42] Q.Wang, J.Peng, H.Kan: Constructions of Cryptographically Significant Boolean functions Using Primitive Polynomials. IEEE Trans. on Inform. Th., Vol. 56, no. 6, June 2010.

[43] D.Yusong, P.Dingyi: Construction of Boolean functions with maximum algebraic immunity and count of their annihilators at lowest degree. Science China, Information Sciences 53(4): 780-787 (2010).

[44] X.M.Zhang, Y.Zheng: Cryptographically Resilient Functions. IEEE Trans. on Inform. Th., Vol. 43, no. 5, September 1997.

[45] X.Zeng, C.Carlet, J.Shan, L.Hu: More Balanced Boolean functions With Optimal Algebraic Immunity and Good Nonlinearity and Resistance to Fast Algebraic Attacks. IEEE Trans. on Inform. Th. 57(9): 6310-6320 (2011).

[46] F.Zhang, Y.Hu, H.Ma, M.Xie: Constructions of Maiorana-McFarland's Bent Functions of Prescribed Degree. Proceedings of International Conference on Computational Intelligence and Security (2010). DOI 10.1109/CIS.2010.157.

Appendix A

SOME RESULTS FOR n = 9

Table A.1: Comparison of non-linearities of functions

Defining Polynomial (Integer value)	**Primitive element (Integer value)**	**Non-linearity of function in [11]**	**Elements swapped (root↔support)**	**Non-linearity of improved function**
529	23	232	$\alpha^{1} \leftrightarrow \alpha^{470}$	234
539	10	234	$\alpha^{96} \leftrightarrow \alpha^{355}$	236
647	2	234	$\alpha^{220} \leftrightarrow \alpha^{417}$	236
661	17	234	$\alpha^{152} \leftrightarrow \alpha^{335}$	236
731	64	232	$\alpha^{254} \leftrightarrow \alpha^{505}$	234
847	32	234	$\alpha^{27} \leftrightarrow \alpha^{494}$	236
859	197	232	$\alpha^{182} \leftrightarrow \alpha^{441}$	234
949	219	232	$\alpha^{5} \leftrightarrow \alpha^{260}$	234

Table A.2: Different improved functions from same parent function

Defining Polynomial (Integer value)	**Primitive element (Integer value)**	**Non-linearity of function in [11]**	**Different options for swapping elements**	**Non-linearity of improved function**
529	3	232	(i) $\alpha^{1} \leftrightarrow \alpha^{470}$	234
			(ii) $\alpha^{201} \leftrightarrow \alpha^{343}$	234
			(iii) $\alpha^{5} \leftrightarrow \alpha^{470}$	234
731	64	232	(i) $\alpha^{254} \leftrightarrow \alpha^{505}$	234
			(ii) $\alpha^{184} \leftrightarrow \alpha^{506}$	234
			(iii) $\alpha^{249} \leftrightarrow \alpha^{506}$	234
901	386	232	(i) $\alpha^{241} \leftrightarrow \alpha^{261}$	234
			(ii) $\alpha^{254} \leftrightarrow \alpha^{259}$	234
			(iii) $\alpha^{144} \leftrightarrow \alpha^{325}$	234

Appendix B

SOME RESULTS FOR n = 10

Table B.1: Comparison of non-linearities of functions

Defining Polynomial (Integer value)	**Primitive element (Integer value)**	**Non-linearity of function in [11]**	**Elements swapped (root↔support)**	**Non-linearity of improved function**
1033	2	478	$\alpha^3 \leftrightarrow \alpha^{571}$	480
1051	16	480	$\alpha^{12} \leftrightarrow \alpha^{856}$	482
1163	399	480	$\alpha^{367} \leftrightarrow \alpha^{1007}$	482
1239	2	478	$\alpha^{92} \leftrightarrow \alpha^{945}$	480
1305	903	478	$\alpha^{72} \leftrightarrow \alpha^{815}$	480
1347	32	478	$\alpha^{205} \leftrightarrow \alpha^{863}$	480
1431	4	478	$\alpha^{392} \leftrightarrow \alpha^{582}$	480
2011	16	482	$\alpha^{1} \leftrightarrow \alpha^{630}$	484

Table B.2: Different improved functions from same parent function

Defining Polynomial (Integer value)	**Primitive element (Integer value)**	**Non-linearity of function in [11]**	**Different options for swapping elements**	**Non-linearity of improved function**
1033	2	478	(i) $\alpha^3 \leftrightarrow \alpha^{571}$	480
			(ii) $\alpha^5 \leftrightarrow \alpha^{761}$	480
			(iii) $\alpha^6 \leftrightarrow \alpha^{853}$	480
1305	903	478	(i) $\alpha^{72} \leftrightarrow \alpha^{815}$	480
			(ii) $\alpha^{170} \leftrightarrow \alpha^{793}$	480
			(iii) $\alpha^{202} \leftrightarrow \alpha^{642}$	480
1431	4	478	(i) $\alpha^{392} \leftrightarrow \alpha^{582}$	480
			(ii) $\alpha^{501} \leftrightarrow \alpha^{749}$	480
			(iii) $\alpha^{27} \leftrightarrow \alpha^{590}$	480

Appendix C

SOME RESULTS FOR n = 11

Table C.1: Comparison of non-linearities of functions

Defining Polynomial (Integer value)	**Primitive element (Integer value)**	**Non-linearity of function in [11]**	**Elements swapped (root↔support)**	**Non-linearity of improved function**
2053	6	982	$\alpha^{631} \leftrightarrow \alpha^{1584}$	984
2071	2044	982	$\alpha^{467} \leftrightarrow \alpha^{1886}$	984
2119	7	980	$\alpha^{545} \leftrightarrow \alpha^{1352}$	982
2147	16	982	$\alpha^{430} \leftrightarrow \alpha^{1531}$	984
2421	2	982	$\alpha^{1} \leftrightarrow \alpha^{1122}$	984
2955	7	986	$\alpha^{529} \leftrightarrow \alpha^{1375}$	988
3573	596	986	$\alpha^{141} \leftrightarrow \alpha^{1388}$	988
3851	746	982	$\alpha^{473} \leftrightarrow \alpha^{1057}$	984

Table C.2: Different improved functions from same parent function

Defining Polynomial (Integer value)	**Primitive element (Integer value)**	**Non-linearity of function in [11]**	**Different options for swapping elements**	**Non-linearity of improved function**
2119	7	980	(i) $\alpha^{545} \leftrightarrow \alpha^{1352}$	982
			(ii) $\alpha^{316} \leftrightarrow \alpha^{1764}$	982
			(iii) $\alpha^{162} \leftrightarrow \alpha^{1862}$	982
2421	2	982	(i) $\alpha^{1} \leftrightarrow \alpha^{1122}$	984
			(ii) $\alpha^{501} \leftrightarrow \alpha^{1025}$	984
			(iii) $\alpha^{985} \leftrightarrow \alpha^{1253}$	984
3851	746	982	(i) $\alpha^{473} \leftrightarrow \alpha^{1057}$	984
			(ii) $\alpha^{650} \leftrightarrow \alpha^{1677}$	984
			(iii) $\alpha^{857} \leftrightarrow \alpha^{2012}$	984

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Ahmed Khan, Mansoor
Nationality: Pakistani
Date and Place of Birth: 01-09-1976, Karachi, Pakistan
Marital Status: Married
Phone: +905412525428

EDUCATION

Degree	Institution	Year of Graduation
M.S. (Information Security)	MCS, NUST, Islamabad, Pakistan	2008
B.S. (Avionics Engineering)	CAE, NUST, Risalpur, Pakistan	1997
High School (Science)	PEA, Dubai, UAE	1993

PROFESSIONAL EXPERIENCE

Year	Place	Position
1994 till date	Pakistan Air Force	Avionics Engineering Officer

INTERNATIONAL CONFERENCE PUBLICATIONS

1. *"Pseudo Random Number Based authentication to counter denial of service attacks on 802.11"*; IEEE Explore in 5th IFIP International Conference on Wireless and Optical Communications Networks, 2008. WOCN '08.

2. *"Improved Nonce Construction Scheme for AES CCMP to Evade Initial Counter Prediction"*; IEEE Explore in Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/ Distributed Computing, 2008. SNPD '08.

3. *"Improvement in Non-linearity of Carlet-Feng Infinite Class of Boolean Functions"*; Cryptology and Network Security. CANS '12. Lecture Notes in Computer Science (LNCS) Volume 7712, 2012, pp 280-295.