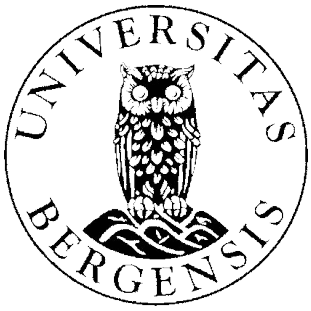
University of Bergen
Faculty of Mathematical and Natural Sciences
Department of Informatics
The Selmer Center



A DATABASE FOR BOOLEAN FUNCTIONS AND CONSTRUCTIONS OF GENERALIZED COMPLEMENTARY PAIRS

by

Mohamed Ahmed A. M. A. Abdelraheem

A thesis report submitted in partial fulfillment of the requirements for the degree of
Master of Science in Informatics

June 2008

To my

mother, father and brothers

with love

Abstract

In this thesis, we study spectral measures of Boolean functions. In the first half of thesis, we study the Walsh spectrum and the periodic autocorrelation spectrum of a Boolean function. A database of Boolean functions is implemented and described, and a survey is presented of cryptographic criteria, most of which are included within the database. In the second half of the thesis, we study the aperiodic autocorrelation spectrum of a Boolean function and some more spectral measures with respect to certain types of unitary matrix. We investigate the Turyn construction for Golay complementary pairs. We show how to convert this construction so as to realize three distinct types of complementary construction. We focus, in particular, on the construction of Boolean function pairs which are Type-I, Type-II or Type-III complementary or near-complementary.

Acknowledgements

First and foremost, I would like to thank my supervisor M. Parker for showing me how scientific research is done. Thank you for showing me how theorems and lemmas are developed. You have been a wonderful supervisor. I am very grateful to you. I benefited a lot from your supervision and really owe you more than you know. Words fail to express my appreciation to your supervision.

I would like to to express my gratitude to Sondre Ronjom and Lars Erik Danielsen for letting me work on their own Boolean function database website. In fact, I did nothing than adding few things to their valuable work.

Where would this thesis be without LaTeX? Many thanks go to my friend M. Ali for showing me the way to LaTeX my thesis.

I would like to thank the Norwegian State Educational Loan Fund for granting me a scholarship throughout my master studies. Thank you for giving me the chance to study in Norway.

Last but not least, I would like to thank my family for their constant support and prayers.

Table of Contents

Chapter 1

Introduction

Boolean functions are functions from the vector space of all binary vectors of length n, ${F_2}^n$, to the finite field F_2 ($\{0, 1\}$). They play an important role in coding theory and a fundamental role in cryptology. In both applications, Boolean functions with a small number of variables n are used in practice due to efficiency. Though n is currently small, studying and determining those n small Boolean functions with specific and desired properties is a hard problem that cannot be solved by an exhaustive search due to the size of the space of n- variable Boolean functions which is 2^{2^n}. This size is huge for n $\geq$ 6. For instance suppose that we have a computer that performs 10^9 operations per second, then for $n = 6$ we have $2^{2^6} = 2^{64} \approx 10^{19}$ different Boolean functions which means that our computer will spend 10^{10} seconds $\approx$ 31 years looping through all of them. When $n = 7$ it will spend much longer than the current age of the universe. This simply means that, for $n \geq 6$, exhaustive search is infeasible. So looking for desired functions should employ clever computer investigations(heuristic search) or mathematical constructions(algebraic techniques) or employ a combination of both investigations and algebraic techniques.

In this thesis, we implement a Boolean function database website that contains desirable Boolean functions found by researchers throughout the world. The website calculates certain properties of a Boolean function, and encourages the user to save the Boolean function to the website if it is a good one, and retrieves the Boolean function according to the conditions entered by the user. The website displays the bounds on the properties of the Boolean functions in the database. It also calculates theoretical bounds on the properties of a Boolean function.

Conventionally, researchers identify periodic cryptographic criteria for a Boolean function and these are what we focus on in the Boolean functions database. In contrast, in the second half of the thesis, we consider the aperiodic autocorrelation spectrum of a Boolean function and some more spectral measures with respect to certain types of unitary matrix. Specifically, we investigate the Turyn construction for Golay complementary pairs. We show how to convert this construction so as to realize three distinct types of complementary construction. We focus, in particular, on the construction of Boolean function pairs which are Type-I, Type-II or Type-III complementary or near-complementary.

The rest of the thesis is organized as follows:

In Chapter 2, we begin by giving a brief introduction to cryptology and the use of Boolean functions in cryptology. Then we discuss three different ways to represent Boolean functions.

In Chapter 3, we begin by defining two useful analytic tools, the Walsh Spectrum and the periodic autocorrelation spectrum, which are used in describing many of the cryptographic properties of Boolean functions. After this, we discuss the properties that are related to the Walsh spectrum and the periodic autocorrelation spectrum. We close the Chapter by discussing other properties that are not directly related to Walsh Spectrum and the periodic autocorrelation spectrum.

Chapter 4 is devoted to theoretical bounds on the properties of Boolean functions. We discuss many of the currently known bounds on the algebraic degree, nonlinearity, correlation immunity, resilience, propagation criteria, absolute indicator and sum of squares indicator.

Chapter 5 describes the implemented Boolean Functions Database Website and lists the objectives of this Website.

Chapter 6 surveys Golay complementary sequences and arrays. It discusses the existence of Golay sequences and arrays, describes the spectral properties of Golay sequences and arrays, and also describes some of the standard constructions of Golay sequences and arrays.

In Chapter 7, we introduce Type-I, Type-II and Type-III complementary array pairs.

We then discuss the constructions of Type-I, Type-II and Type-III array pairs and the conversions among Type-I, Type-II and Type-III. We also present binary constructions for Type-I, Type-II and Type-III complementary array pairs, where each array dimension is of length 2 and characterize all known binary pairs in Type-I, Type-II and Type-III that could be constructed by recursively using these binary constructions. We close the chapter by using these binary constructions to find near-complementary binary pairs in Type-I and Type-II simultaneously.

Chapter 8 wraps up what has been accomplished in this thesis and presents some future work.

Chapter 2

Introduction to Boolean functions

Cryptology is the study that embodies cryptography and cryptanalysis. *Cryptography* is the study of designing cryptosystems, while *cryptanalysis* is the study of breaking these cryptosystems. The main objective of cryptography is to secure the communication between two or more channels by transforming the transmitted message(plaintext) to a message(ciphertext) that cannot be recovered by an adversary to its original status before the transformation. The transformation from plaintext to ciphertext is called encryption and the recovery of the plaintext from the ciphertext is called decryption.

Encryption-decryption cryptography is the classical cryptography. Modern cryptography has embodied other techniques such as authentication, data integrity and non-repudiation [23]. The study of encryption-decryption can be divided into symmetric cryptography and public key cryptography. Symmetric cryptography is the process of encrypting and decrypting a message using the same key, while public key cryptography is the process of encrypting a message by a public key and decrypting it by a private key. The study of symmetric cryptography includes the study of stream ciphers and block ciphers and their applications. Since we focus on Boolean functions in this thesis, we will explain stream and block ciphers a little bit more.

2.1 Stream Ciphers

A stream cipher operates on individual bits. The provably secure stream cipher, called the *one time pad*, is a stream cipher whose secret key has the same length as the plaintext. The cipher xors the secret key bits with plain text bits. Modern stream ciphers try to embody

the one time pad by using a short key to generate a much longer pseudo-random key stream sequence. Constructing robust stream ciphers requires keystreams of long period otherwise the stream cipher will be vulnerable to certain attacks.

Linear Feedback Shift Registers(LFSRs) devices are used to produce long period sequences from short ones. An LFSR of length n consists of n stages. Each stage stores one bit(or word) and has one input and one output. The flow of bits is controlled by a clock. At each clock tick, the contents of stage 0 is the output and forms part of the output sequence. Stage i is moved to stage $i-1$ for each $1 \leq i \leq n-1$. Stage $n-1$ is filled by the feedback bit(or word), s_j, which is formed by xoring together a fixed subset of the previous stages $(0, 1, \ldots, n-1)$ depending on the structure of the LFSR (see the following figure, $s_j = \bigoplus_{i=1}^{n} c_i s_{j-i}$) [23]. LFSRs are vulnerable to the powerful $O(n^2)$ linear com-

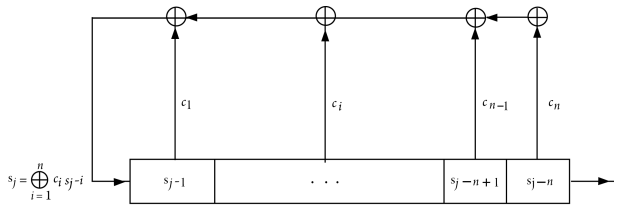


Figure 2.1: LFSR

plexity Berlekamp-Massey attack [23] which requires only $2n$ consecutive sequence bits(or words) to deduce the c_i's. Therefore, the LFSR is used together with a nonlinear Boolean function to avoid the Berlekamp-Massey attack. The most used LFSR models are the filter generator and the combining LFSR.

2.2 Block Ciphers

A block cipher divides the plaintext into block of bits with the same length and then encrypts each block by the secret key. As in stream ciphers, Boolean functions play an important role in block ciphers. Every block cipher takes as input a block of plaintext represented in bits $(x_0, \ldots, x_{n-1})$ and outputs a binary vector $(y_0, \ldots, y_{m-1})$ depending on

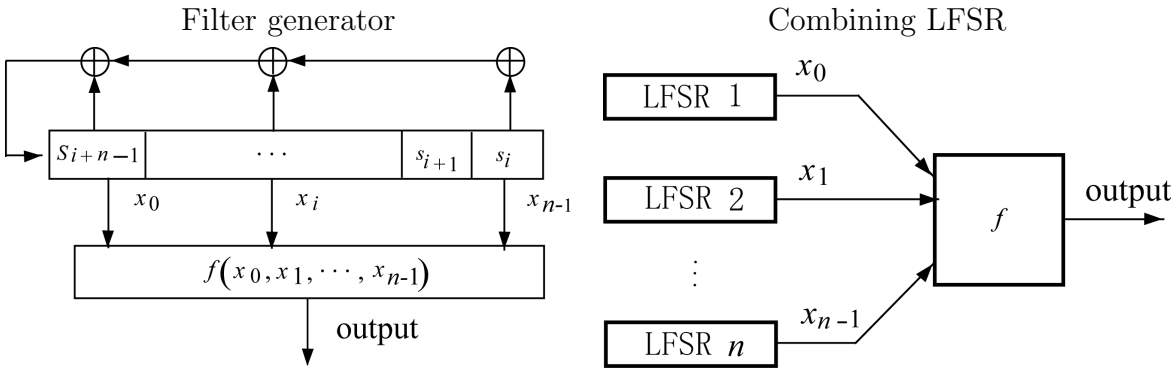


Figure 2.2: Filter generator and Combining LFSR

the secret key. $y_0, \ldots, y_{m-1}$ are the outputs of Boolean functions having $x_0, \ldots, x_{n-1}$ and the secret key as their input parameters.

2.3 Representations of Boolean functions

There are many ways to represent a Boolean function. The advantage of each representation over the other depends on the application that is using the Boolean function. We present now three different representations.

Truth Table(TT)

The truth table representation is the default representation of a Boolean function as it directly translates the definition of a Boolean function. The TT of a Boolean function f on ${F_2}^n$ is a binary vector of length 2^n, each element of this binary vector is an image corresponding to a unique element in ${F_2}^n$. Now we introduce a notation that lets us order the elements of the TT lexicographically. We replace each element $(x_0, x_1, \ldots, x_{n-1})$ in ${F_2}^n$ by its decimal representation $x = x_0 2^{n-1} + x_1 2^{n-2} + ... + x_{n-1}$. So instead of writing $f(0, 0, \ldots, 0)$ we write $f(0)$, instead of $f(0, 0, \ldots, 1)$ we write $f(1)$, instead of $f(1, 1, \ldots, 1)$ we write $f(2^n - 1)$ and so on. This gives a lexicographical order on all the elements of

${F_2}^n$ and allows us to define a Boolean function as $f = [f(0)\ f(1)\ f(2)\ \dots f(2^n - 1)]$. For instance, suppose we have a 3-variable Boolean function $f = [0\ 1\ 1\ 0\ 0\ 1\ 0\ 1]$. Then our TT will be as shown in Table 2.1.

x	x_0	x_1	x_2	$f(x)$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Table 2.1: Truth table

Another representation that is closely related to the truth table is the polarity TT(PTT) or bipolar representation, and is widely used in telecommunications. It is defined as $(-1)^f = [(-1)^{f(0)}\ (-1)^{f(1)}\ \dots\ (-1)^{f(2^n-1)}]$ which means that instead of 0's in TT we have 1's in the PTT and instead of 1's in TT we have -1's in the PTT. So it is a sequence of $\{1, -1\}$'s.

Algebraic Normal Form(ANF)

The ANF is one of the most used representations in cryptography. An ANF of a Boolean function on ${F_2}^n$ is a polynomial of the following form:

$$f(x_0, x_1, \dots, x_{n-1}) = \sum_{j=(j_0,\dots,j_{n-1})\in F_2^n} a_j x_0^{j_0} x_1^{j_1} \cdots x_{n-1}^{j_{n-1}} \ (mod\ 2)$$

where $a_j \in F_2$.

The *algebraic degree of* f, denoted by *deg(f)*, is the number of variables in the longest term(s) of the ANF of f. If *deg(f)* ≤ 1, then f is called an *affine* function. An affine

function without the constant term (*i.e.* $a_0 = 0$) is often called a *linear* function. An affine function with *deg(f)* $= 0$, which is either $f(x) = 0$ or $f(x) = 1$, is called a *constant* function. The set of affine functions is denoted by $A(n)$.

Let $C = [c_0\ c_1\ \ldots\ c_{2^n-1}]$ be the coefficient vector of the polynomial representing the Boolean function f. If $c_j = 1$, where $0 \leq j \leq 2^n-1$, then the monomial $x_0^{j_0}x_1^{j_1}\cdots x_{n-1}^{j_{n-1}}$ exists in the ANF of f and does not otherwise, where $(j_0, j_1, \ldots, j_{n-1})$ is the binary representation of index j. For instance, if $C = [0\ 0\ 0\ 0\ 1\ 0\ 0\ 1]$ then the ANF is $x_0 + x_0x_1x_2$. The following theorem, shows a relation between C and the truth table $f = [f(0)\ f(1)\ f(2)\ \ldots\ f(2^n-1)]$.

Theorem 1. *[32] Let f be the truth table of an n-variable Boolean function. Let be as defined above. Then*

$$C = fA_n$$

where

$$A_n = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{\otimes n}$$

That is A_n is the nth tensor power[1] *mod 2 of the matrix* $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$*, or in other notation,*

$$A_n = \begin{bmatrix} A_{n-1} & A_{n-1} \\ 0 & A_{n-1} \end{bmatrix} \qquad \textit{and} \qquad A_0 = [1]$$

Proof:

We prove the theorem by induction on n. Let $n = 1$. Take any Boolean function in one

[1]The tensor product of a $p \times q$ matrix U and a $k \times l$ matrix V is defined by the $pk \times ql$ matrix $U \otimes V = \begin{bmatrix} u_{11}V & u_{12}V & \ldots \\ u_{21}V & u_{22}V & \ldots \\ \vdots & \vdots & \ddots \end{bmatrix}$, where u_{ij} is the entry of row i and column j in U. The nth tensor power of a matrix U is the tensor product of U with itself n times: $U \otimes U \otimes \cdots \otimes U$.

variable $f = [f(0)\ f(1)]$ and multiply it by

$$A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

We get $[f(0)\ f(0)+f(1)]$. On the other hand it is obvious that $f(x_0) = f(0)+(f(0)+f(1))x_0$ which indicates that $C = [f(0)\ f(0) + f(1)]= fA_1$. So the theorem is true when n = 1. To clarify the induction step, we again verify the theorem when n = 2, $f = [f(0)\ f(1)\ f(2)\ f(3)] = [f(0,0)\ f(0,1)\ f(1,0)\ f(1,1)]$. Multiplying f by A_2, we get $fA_2 = [f(0,0)\ f(0,0) + f(0,1)\ f(0,0) + f(1,0)\ f(0,0) + f(0,1) + f(1,0) + f(1,1)]$. On the other hand

$$f(x_0, x_1) = f(0, x_1) + (f(0, x_1) + f(1, x_1))x_0$$

Substituting $f(0, x_1)$ by $f(0,0) + (f(0,0) + f(0,1))x_1$ and $f(1, x_1)$ by $f(1,0) + (f(1,0) + f(1,1))x_1$, we get $f(x_0, x_1) = f(0,0)+(f(0,0)+f(0,1))x_1+(f(0,0)+f(1,0))x_1+(f(0,0)+f(0,1) + f(1,0) + f(1,1))x_0x_1$, we see that $C = [f(0,0)\ f(0,0) + f(0,1)\ f(0,0) + f(1,0)\ f(0,0) + f(0,1) + f(1,0) + f(1,1)] = fA_2$ and this proves that the theorem holds for $n = 2$.

Now we assume that the theorem holds when the number of variables is less than n. We want to prove that the theorem is true when the number of variables is n. Let $f(0, x_1, \ldots, x_{n-1}) = f_0(x_0, \ldots, x_{n-1})$ and $f(1, x_1, \ldots, x_{n-1}) = f_1(x_0, \ldots, x_{n-1})$. Obviously, $f_0 = [f_0(0) \ldots f_0(2^{n-1})]$, $f_1 = [f_1(0) \ldots f_1(2^{n-1})]$ and $f = [f_0\ f_1]$. Let C_0 be the coefficient vector related to f_0 and C_1 be the coefficient vector related to f_1. Then by the induction hypothesis $C_0 = f_0A_{n-1}$ and $C_1 = f_1\ A_{n-1}$. Now f=f_0+$(f_0$+$f_1)x_0$, has coefficient vector $C = [C_0\ C_0 + C_1]$, substituting C_0 by f_0A_{n-1} and C_1 by f_1A_{n-1}, we find that $C = [f_0A_{n-1}\ f_0A_{n-1} + f_1A_{n-1}] = fA_n$, which proves that the theorem holds when the number of variables is n.□

Theorem 1 helps us to convert from truth table to ANF and vice versa in almost 2^{2n} binary operations. The following algorithm reduces the conversion to only $O(n2^n)$ operations.

Algorithm 1:

Input: TT of a Boolean function f

Output: The coefficient vector of the ANF of f

For $0 \leq k \leq n$, define $f_{k,a} \in F_{2^k}$, where $0 \leq a \leq 2^{n-k} - 1$.

1. Set $f_0, a = f(a)$ for $0 \leq a \leq 2^n - 1$.

2. *for* $k = 0$ to $n - 1$ do

   for b $= 0$ to $2^{n-k-1} - 1$ do

   $f_{(k+1),b} = [f_{k,2b}\ f_{k,(2b+1)} + f_{k,2b}]$

3. $C = f_{n,0}$

The following example illustrates what the above algorithm does. Let us find the ANF of the Boolean function represented by the truth table in Table 2.1. We have $f = [0\ 1\ 1\ 0\ 0\ 1\ 0\ 1]$. Looking at the 1's positions in C we see that the ANF of f is $x_2 + x_1 + x_0x_1$.

f	0	1	1	0	0	1	0	1
$k = 0$	0	1	1	1	0	1	0	1
$k = 1$	0	1	1	0	0	1	0	0
$k = 2$	0	1	1	0	0	0	1	0
$C = f_{3,0}$	0	1	1	0	0	0	1	0

Table 2.2: Converting TT to ANF algorithm

Trace representation

The trace representation plays an important role in sequence theory, and is also used for defining and studying Boolean functions [4]. In the theory of finite fields, the trace function on the finite field F_{p^n} is the function $Tr\colon F_{p^n} \to F_p$ defined by $Tr(x) = x + x^p + x^{p^2} + x^{p^3} + \ldots + x^{p^{n-1}}$. Here we are considering the case when p = 2, that is, when our finite field is the binary field F_{2^n}. So our trace is a function $Tr\colon F_{2^n} \to F_2$. Define the function $Tr(\sum_{i=0}^{i=k} x^{a_i t + b_i})$ on F_{2^n} for $0 \leq t \leq 2^n - 2$ and integers a, b. Let $p(x)$ be a

primitive polynomial over F_{2^n}. Then x can generate F_{2^n}, *i.e.*, x^t where $0 \leq t \leq 2^n - 2$ are all nonzero elements of F_{2^n}. From the theory of finite fields, we know that each element in F_{2^n} can be represented by a binary string of length n, and we also know that F_{2^n} consists of all the possible binary strings of length n. This means that, for each value of x^t, we have a corresponding binary string. By evaluating $Tr(\sum_{i=0}^{i=k} x^{a_i t+b_i})$ for $0 \leq t \leq 2^n - 2$, we obtain $2^n - 1$ binary values. Now, for each t, let $Tr(\sum_{i=0}^{i=k} x^{a_i t+b_i})$ be an element in the truth table at the position corresponding to the decimal representation of the binary string corresponding to the element x^t. Now, if we set a value at position 0, we will have a complete truth table. This value can be either true or false, but by convention we set it as false. The general form of the trace function we are dealing with is $Tr(\sum_{i=0}^{i=k} x^{a_i t+b_i})$. There is a restriction on the values of $b_1, b_2, b_3, \ldots, b_k$ depending on $a_1, a_2, a_3, \ldots, a_k$ respectively. To validate b_i, where $1 \leq i \leq k$ we do the following steps:

1. Compute $v = \gcd(2^n - 1, a_i)$.
2. Compute $u = (2^n - 1)/v$.
3. Find the smallest number r such that u divides $2^r - 1$.
4. Compute $e = (2^n - 1)/(2^r - 1)$.
5. If $x^{b_i} \in \{0\ x^e\ x^{2e}\ \ldots\ x^{(2r-1)e}\}$, then b_i is valid.

Since the trace function is linear then $Tr(\sum_{i=0}^{i=k} x^{a_i t+b_i}) = \sum_{i=0}^{i=k} Tr(x^{a_i t+b_i})$. After validating b_i and a_i, we consider computing the trace function over the finite field F_{2^r} rather than F_{2^n}, where r is as noted above.

Let us take an example to explain how to convert a trace representation to a truth table representation. Suppose we have a Boolean function represented by $Tr(x^{3t+2})$ with a primitive polynomial $p(x) = x^3 + x + 1$ on the finite field F_{2^3}. The finite field F_{2^3} indicates that the Boolean function represented by $Tr(x^{3t+2})$ is a 3 variable Boolean function. Computing $Tr(x^{3t+2})$ for $0 \leq t \leq 2^3 - 2 = 6$, gives us the following table,

α	x_0	x_1	x_2	$Tr(\alpha)$
1	0	0	1	0
x	0	1	0	1
x^2	1	0	0	0
x^3	0	1	1	0
x^4	1	1	0	1
x^5	1	1	1	1
x^6	1	0	1	1

Table 2.3: Trace computation

Permuting the table to be in a lexicographical order and adding the all zero row, we get the following truth table

α	x_0	x_1	x_2	$Tr(\alpha)$
0	0	0	0	0
1	0	0	1	0
x	0	1	0	1
x^3	0	1	1	0
x^2	1	0	0	0
x^6	1	0	1	1
x^4	1	1	0	1
x^5	1	1	1	1

Table 2.4: Truth table representation of $Tr(x^{3t+2})$

Let us see how the trace representation can be obtained from the truth table of a Boolean function. By ordering the truth table according to the generator of the finite field F_{2^n} and removing the all zeros entry we get a binary vector of length $2^n - 1$. Using the inverse of the Galois discrete Fourier transform on this binary vector, we get a vector V, with entries

in finite field F_{2^n}. This vector has properties that lead us to deduce the trace function. Let c_i be the coset leader of the cyclotomic coset i of F_{2^n}, where $1 \leq c_i \leq 2^n - 2$. If the entry at position c_i in the vector V is nonzero then all the entries at the positions corresponding to the elements of coset i should be nonzero. If x^j is an entry in the vector V at position c_i, then the corresponding trace formula is $Tr(x^{c_i t+j})$. Thus for each coset leader, we have a corresponding trace formula. So we see that the number of trace formulas depends on the number of nonzero elements of the vector V. The sum of these trace formulas is the trace function corresponding to the TT of the Boolean function. To do this transformation we multiply our vector by the Inverse Galois Discrete Fourier Transform (IGDFT) matrix which is an $2^{n-1} \times 2^{n-1}$ matrix,

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha^{1\cdot 1} & \alpha^{1\cdot 2} & \cdots & \alpha^{1\cdot 2^{n-1}} \\ 1 & \alpha^{2\cdot 1} & \alpha^{2\cdot 2} & \cdots & \alpha^{2\cdot 2^{n-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{2^{n-1}\cdot 1} & \alpha^{2^{n-1}\cdot 2} & \cdots & \alpha^{2^{n-1}\cdot 2^{n-1}} \end{bmatrix}$$

where $\alpha = x^{2^n-2}$ and x is a generator of F_{2^n} according to some primitive polynomial $p(x)$ in F_{2^n}.

The following example demonstrates how to convert a truth table Boolean function to a trace Boolean function. Consider the Boolean function represented by Table 2.4. By permuting Table 2.4 back to the table on Table 2.3 and removing the all zeros entry, we get the following binary vector [0 1 1 0 1 0 1]. Now we want the trace function of this binary vector. Computing the inverse discrete Fourier transform on this vector gives us a vector with entries in F_2^3. To compute this transform we multiply our vector by the following

matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \\ 1 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha & \alpha^4 \\ 1 & \alpha^4 & \alpha & \alpha^5 & \alpha^2 & \alpha^6 & \alpha^3 \\ 1 & \alpha^5 & \alpha^3 & \alpha & \alpha^6 & \alpha^4 & \alpha^2 \\ 1 & \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha \end{bmatrix}$$

where $\alpha = x^6$ and $x^3 = x+1$ according to our primitive polynomial $p(x) = x^3+x+1$. This transformation gives us the following vector [0 0 0 x^2 0 x x^4]. Now, we have nonzero elements at positions 3, 5 and 6. These positions are the elements of a cylcotomic coset in F_2^3. We see that 3 is the coset leader, so $a_1 = c_1 = 3$. We also see that x^2 is the element at index c_2, so $b_1 = 2$. Therefore, our trace formula is $Tr(x^{3t+2})$. Since 3, 5 and 6 are the only nonzero positions then there is only one trace formula. Suppose that we get the following vector [0 x^4 x x^2 x^2 x x^4] in some transformation. The nonzero elements give us the following cyclotomic cosets in F_2^3. We see that we have two cyclotomic cosets, {1, 2, 4} and {3, 5, 6}. They give us two trace formulas, $Tr(x^{t+4})$ and $Tr(x^{3t+2})$. The sum of them, $Tr(x^{t+4}) + Tr(x^{3t+2})$, gives us the corresponding trace function.

Chapter 3

Boolean functions cryptographic criteria

In his seminal paper [38], Claude Shannon suggested two statistical properties that any classical cryptosystem should possess in order to be secure against statistical analysis, namely *confusion* and *diffusion*. *Confusion* means complicating the relation between the key and the ciphertext in a way such that each ciphertext bit depends on several bits of the secret key. *Diffusion* means distributing the secret key over the plaintext in a way such that each key bit affects as many bits as possible in the ciphertext. In other words, diffusion is enhanced if changing of a key bit in the plaintext changes several bits in the ciphertext. Most of the attacks on classical cryptosystems exploit weaknesses of these two important properties in the system under attack.

Attacks on classical cryptosystems have led to criteria that must be satisfied by cryptographic Boolean functions embedded in those cryptosystems. To name a few, nonlinearity and correlation immunity are proposed criteria for a Boolean function to resist an affine approximation attack on the filter generator and a correlation attack on the combining generator respectively. They both achieve confusion by complicating the relation between the ciphertext and the keystream.

Before going deep into the criteria of Boolean functions, we need to introduce some definitions that will be used later. The Hamming weight of a binary vector $u \in {F_2}^n$, denoted by $wt(u)$, is the number of nonzero places in u. Since a Boolean function is a binary vector, the Hamming weight of a Boolean function f on ${F_2}^n$, denoted by $wt(f)$, is the number of nonzero places or the number of 1's in f. The Hamming distance between two Boolean

functions f_1 and f_2, denoted by $d(f_1, f_2)$, is the size of the set $\{x \in {F_2}^n : f_1(x) \neq f_2(x)\}$ which is equal to $wt(f_1 + f_2)$.

3.1 Analytic tools for Boolean functions

The discrete Fourier transform of a Boolean function $f(x)$ is defined to be the real valued function $\hat{f}(a)$ defined on ${F_2}^n$,

$$\hat{f}(a) = \sum_{x \in {F_2}^n} f(x)(-1)^{a \cdot x} \tag{3.1}$$

where $a \cdot x = a_0 x_0 + \cdots + a_{n-1} x_{n-1}$. Note that $\hat{f}(\mathbf{0})$ equals the Hamming weight of f.
The transform of the sign function of f, defined by $(-1)^f$, equals,

$$W_f(a) = \sum_{x \in {F_2}^n} (-1)^{f(x)+a \cdot x} \tag{3.2}$$

This transform is called the *Walsh transform*. It is easy to see that $W_f(a) = 2^n - wt(f(x) + a \cdot x)$.
Substituting $(-1)^{f(x)}$ by $1 - 2f$, we find a relation between $W_f(a)$ and $\hat{f}(a)$,

$$W_f(a) = \begin{cases} -2\hat{f}(a) & \text{if } a \neq \mathbf{0} \\ 2^n - 2\hat{f}(a) & \text{if } a = \mathbf{0} \end{cases}$$

The *derivative* of f with respect to a vector $b \in F_2^n$ is the Boolean function $D_d(f) = f(x) + f(x + d)$. The *periodic autocorrelation function* of f is a real-valued function defined on all $a \in F_2^n$

$$\Delta_f(a) = \sum_{x \in {F_2}^n} (-1)^{f(x)+f(a+x)} = \sum_{x \in {F_2}^n} (-1)^{D_a(f)} \tag{3.3}$$

The following theorem shows the expression of the periodic autocorrelation function $\Delta_f(a)$ for all $a \in F_2^n$ in terms of the Walsh transform.

Theorem 2. *Wiener-Khintchine:* $\Delta_f(a) = \sum_{u \in {F_2}^n} W_f(u)^2 (-1)^{a.u}$.

In this chapter, we will see the importance of the Walsh transform and the periodic autocorrelation in the study of Boolean functions.

3.2 Criteria related to Walsh transform

The affine approximation attack [22, 43] on the filter generator chooses the best affine approximation of the filter generator Boolean function. To choose the best affine approximation, the attack calculates $Pr(f(x) = a \cdot x + b)$ for all $a \in {F_2}^n$ and $b \in F_2$ and chooses the affine function $a \cdot x + b$ with the highest probability. In order to better understand the criteria related to Walsh transform, we need to calculate this probability[1].

Let N_0 denote the number of $x = u$ such that $f(u) = a.u$ and N_1 denote the number of $x = u$ such that $f(u) = a.u + 1$. Obviously, $N_0 + N_1 = 2^n$. Let p_a denotes $Pr(f(x) = a \cdot x)$ and q_a denotes $Pr(f(x) = a \cdot x + 1)$. We see that $p_a = \frac{N_0}{2^n}$ and $q_a = \frac{N_1}{2^n} = 1 - \frac{N_0}{2^n}$. Looking back at equation 3.2, the Walsh transform can be written in terms of N_0 and N_1,

$$W_f(a) = \sum_{x \in {F_2}^n} (-1)^{f(x)+a \cdot x} = N_0 - N_1 \tag{3.4}$$

and this is because

$$(-1)^{f(x)+a \cdot x} = \begin{cases} 1 & \text{if } f(x) = a \cdot x \\ -1 & \text{if } f(x) = a \cdot x + 1 \end{cases}$$

Replacing N_1 by $2^n - N_0$ in equation 3.4, gives us

$$W_f(a) = 2N_0 - 2^n = 2^{n+1} p_a - 2^n = 2^{n+1}(p_a - \frac{1}{2}) \tag{3.5}$$

Rearranging equation 3.5, we find the following formulas for p_a and q_a

$$p_a = \frac{1}{2} + \frac{1}{2^{n+1}} W_f(a) \tag{3.6}$$

$$q_a = \frac{1}{2} - \frac{1}{2^{n+1}} W_f(a) \tag{3.7}$$

[1]The calculations that yield equations 3.4, 3.5, 3.6, an 3.7 are quoted from [32].

Equation 3.6 states the probability that the linear function $a \cdot x$ is a good approximation for $f(x)$ if the Walsh transform of f with respect to a, $W_f(a)$, has a big value. The bigger $W_f(a)$ is, the better $a \cdot x$ is in approximating $f(x)$. Equation 3.7 states that the affine function $a \cdot x + 1$, which is the complement of the linear function $a \cdot x$, is a good approximation for $f(x)$ if $W_f(a)$ is a small negative number. To sum up, whenever the absolute value of $W_f(a)$, $|W_f(a)|$, is big, we have a good approximation whether of the form $a \cdot x$ or $a \cdot x + 1$. The maximum absolute value of the Walsh spectrum gives us the best approximation to $f(x)$ with probability $\frac{1}{2} + (-1)^{\psi(\max_{a \in F_2^n} |W_f(a)|)} \frac{1}{2^{n+1}} \max_{a \in F_2^n} |W_f(a)|$, where $\psi(\max_{a \in F_2^n} |W_f(a)|)$ equals 0 when the sign of $\max_{a \in F_2^n} |W_f(a)|$ before taking the absolute value is positive and equals 1 when the sign of $\max_{a \in F_2^n} |W_f(a)|$ before taking the absolute value is negative.

To calculate p_a and q_a, we need to calculate $W_f(a)$. We see that $W_f(a)$ is actually a multiplication of the row vector $(-1)^{f(x)}$ and the column vector $(-1)^{a \cdot x}$. So multiplying the row vector $(-1)^{f(x)}$ by a matrix, name it H_n, gives us the whole Walsh spectrum of the Boolean function f, $W_f = (-1)^f H_n$. The columns of H_n are the column vectors $(-1)^{a_i \cdot x}$ where $0 \leq i \leq 2^n - 1$ and a_i corresponds directly to the binary vector of the binary representation of the decimal integer i and the column vector is ordered lexicographically with respect to the set of $\{x : x \in F_2^n\}$. The matrix $H_n = [(-1)^{a \cdot x}]$ is a $2^n \times 2^n$ with column i represented by the column vector $(-1)^{a_{i-1} \cdot x}$. This matrix is exactly a Hadamard matrix[2]. For instance when $n = 1$, we have $H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and for $n = 2$, we have

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

The following Theorem demonstrates how to simply calculate H_n.

[2]A Hadamard matrix is a square matrix whose elements are either +1 or -1 and whose rows are mutually orthogonal.

Theorem 3. *[32]*

$$H_n = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^{\otimes n}$$

proof:

We prove the statement by induction on n. For n = 1 the theorem is satisfied. So assume the theorem is satisfied when the number of variables is less than n. we have $x = (\hat{x}, x_{n-1})$ where $\hat{x} = (x_0, x_1, \ldots, x_{n-2})$, and $a = (\hat{a}, a_{n-1})$ where $\hat{a} = (a_0, a_1, \ldots, a_{n-2})$. From the definition of H_n, we see that

$$H_n = \begin{bmatrix} (-1)^{\hat{a}.\hat{x}+0.0} & (-1)^{\hat{a}.\hat{x}+0.1} \\ (-1)^{\hat{a}.\hat{x}+1.0} & (-1)^{\hat{a}.\hat{x}+1.1} \end{bmatrix} = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes H_{n-1}$$

But by the induction hypothesis, we have

$$H_{n-1} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^{\otimes n-1}$$

This means that

$$H_n = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^{\otimes n}$$

which proves that the theorem holds for every n. □

The vector and matrix multiplication in the Walsh transform computation, almost takes $O(2^{2n})$ bit operations but there is a fast algorithm similar to Algorithm 1 that dramatically reduces the computation to just $O(n2^n)$ bit operations.

Algorithm 2:

Input: Truth table of a Boolean function f

Output: The Walsh transform spectrum W_f

For $0 \leq k \leq n$, define $W_{f_{k,a}} \in F_{2^k}$, where $0 \leq a \leq 2^{n-k} - 1$.

1. Set $W_{f_{0,a}} = (-1)^{f(a)}$ for $0 \leq a \leq 2^n - 1$.

2. *for* $k = 0$ to $n - 1$ do

   for b $= 0$ to $2^{n-k-1} - 1$ do

   $W_{f_{(k+1),b}} = [W_{f_{k,2b}} + W_{f_{k,(2b+1)}} \qquad W_{f_{k,2b}} - W_{f_{k,(2b+1)}}\ ]$

3. $C = W_{f_{n,0}}$

The following example demonstrates algorithm 2 and finds the best affine approximation. Let $f = [0\ 1\ 1\ 1\ 1\ 1\ 0\ 0]$ be a Boolean function. In order to find the best affine approximation we need to compute the Walsh transform. The following table shows the process of computing the Walsh transform of f using algorithm 2 along with vectors p and q whose coordinate i represents the probability of $Pr(f(x) = a \cdot x)$ and $Pr(f(x) = a \cdot x + 1)$ where $a \in F_2^n$ is the binary representation of integer i, respectively for $0 \leq i \leq 2^n - 1$.

$(-1)^f$	1	-1	-1	1	1	1	-1	-1
$k = 0$	0	2	-2	0	-2	0	2	0
$k = 1$	0	0	0	4	0	0	4	0
$k = 2$	0	0	4	4	0	0	4	-4
$W_f = W_{f_{3,0}}$	0	0	4	4	0	0	-4	4
p	0.5	0.5	0.75	0.75	0	0	0.25	0.75
q	0.5	0.5	0.25	0.25	0	0	0.75	0.25

Table 3.1: Walsh Transform Computation + Finding the best approximation

From the above table, we see that the highest probability occurring in p is at coordinates 3, 4 and 7. Coordinate 3 corresponds to the linear function $(0, 1, 1) \cdot (x_0, x_1, x_2) = x_1 + x_2$, coordinate 4 corresponds to the linear function $(1, 0, 0) \cdot (x_0, x_1, x_2) = x_0$ and coordinate 7 corresponds to the linear function $(1, 1, 1) \cdot (x_0, x_1, x_2) = x_0 + x_1 + x_2$. In q, there is only one highest probability which occurred at coordinate 6 and corresponds to the affine function $(1, 1, 0) \cdot (x_0, x_1, x_2) + 1 = x_0 + x_1 + 1$.

Nonlinearity

The nonlinearity of a Boolean function f on F_2^n, denoted by $nl(f)$, is the minimum Hamming distance between f and the set of all affine functions on F_2^n, $A(n)$. In mathematical terms it is

$$nl(f) = \min_{g \in A(n)} (d(f, g)) \tag{3.8}$$

The nonlinearity criterion can be expressed in terms of the Walsh transform: To compute nonlinearity we need to find $d(f, a \cdot x)$ and $d(f, a \cdot x + 1)$ for all the possible linear functions $a \cdot x$, $d(f, a \cdot x) = 2^n - \#\{x : f(x) = a \cdot x\} = 2^n - 2^n p_a = 2^n - 2^{n-1} - \frac{1}{2} W_f(a)$ $= 2^{n-1} - \frac{1}{2} W_f(a)$, while $d(f, a \cdot x + 1) = 2^n - \#\{x : f(x) = a \cdot x + 1\} = 2^n - 2^n q_a =$ $2^n - 2^{n-1} + \frac{1}{2} W_f(a) = 2^{n-1} + \frac{1}{2} W_f(a)$. This suggests that $\min_{a \in F_2^n}(d(f, a \cdot x), d(f, a \cdot x + 1))$ $= 2^{n-1} - \frac{1}{2} W_f(a)$ and therefore the nonlinearity of f is:

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{a \in F_2^n} |W_f(a)| \tag{3.9}$$

Equation 2.10 implies that the smaller $\max_{a \in F_2^n} |W_f(a)|$ is, the better $nl(f)$ we will have. This raises the question how small can $\max_{a \in F_2^n} |W_f(a)|$ be ? The following theorem will help us to answer this question.

Theorem 4. *Parseval's Equation:* $\sum_{a \in F_2^n} W_f(a)^2 = 2^{2n}$.

Proof:[1]

In matrix notation $\sum_{a \in F_2^n} W_f(a)^2 = [W_f][W_f]^t$, but from equation 2.8, we know that $W_f = (-1)^f H_n$, so we have $[W_f][W_f]^t = [(-1)^f H_n][(-1)^f H_n]^t = [(-1)^f H_n][H_n]^t[(-1)^f]^t =$ $[(-1)^f][H_n][H_n]^t[(-1)^f]^t$, but since H_n is a Hadamard matrix then $H_n H_n^t = 2^n I$ (Orthogonal property of the Hadamard matrix H_n), therefore $[W_f][W_f]^t = 2^n[(-1)^f][(-1)^f]^t =$ $2^n.2^n = 2^{2n}$. $\square$

Theorem 4 implies that the mean of $W_f(a)^2$ equals 2^n, which means that, $\max_{a \in F_2^n} W_f(a)^2 \geq$ 2^n, thus $\max_{a \in F_2^n} |W_f(a)| \geq 2^{\frac{n}{2}}$ and this answers the above question. So an n variable Boolean function f, gets the highest or the maximum possible nonlinearity when

$\max_{a \in F_2^n} |W_f(a)| = 2^{\frac{n}{2}}$. This tells us that,

$$nl(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1} \tag{3.10}$$

This upper bound is valid for every Boolean function f on F_2^n, and it is called the *universal nonlinearity bound*. A Boolean function is highly nonlinear if its nonlinearity is close to the universal bound. When the universal bound is achieved, the corresponding Boolean function is called a *bent function*. Bent functions have nice properties but the fact that they are unbalanced (see the definition of balanced functions below) make them undesirable in practice. At the end of this chapter, we will give a brief discussion about bent functions. We will give a detailed discussion about nonlinearity and its relations with other criteria in Chapter 4.

Balancedness

A Boolean function f on F_2^n is *balanced* if $wt(f) = 2^{n-1}$. In terms of Walsh transform a Boolean function is balanced if $W_f(0) = 0$ which statistically means that $Pr(f(x) = 0) = Pr(f(x) = 1) = \frac{1}{2}$. Cryptographic functions are ideally balanced or near balanced in order to prevent cryptanalysts from statistically analysing their output distributions.

Correlation immunity and resilience

Correlation immunity criteria was introduced by Siegenthaler [39] to resist correlation attacks on stream ciphers whose keys are generated by the combining generator depicted in figure 2.2.

An n variable Boolean function $f(x_0, x_1, \ldots, x_{n-1})$ is *correlation immune (CI) of order m* if $Pr(f = 1 | x_{i_1} = c_1, x_{i_2} = c_2, \ldots, x_{i_m} = c_m) = Pr(f = 1)$ for any choice of distinct $i_1, i_2, \ldots, i_m$ from $\{0, 1, \ldots, n-1\}$ and $c_1, \ldots, c_m$ belong to F_2. The following theorem [44] gives an equivalent non-probabilistic characterization on the definition of correlation immunity.

Theorem 5. *Let $f(x_0, x_1, \ldots, x_{n-1})$ be a boolean function on F_2^n and let f_j be any Boolean function obtained by setting $x_{i_1} = c_1, x_{i_2} = c_2, \ldots, x_{i_m} = c_m$ in $f(x_0, x_1, \ldots, x_{n-1})$ for any choice of distinct $i_1, i_2, \ldots, i_m$ from $\{0, 1, \ldots, n-1\}$ and $c_1, c_2, \ldots, c_m$ belong to F_2. Then f is correlation immune if $wt(f_j) = wt(f)/2^m$.*
Proof:

From the definition, we know that $Pr(f_i = 1) = Pr(f = 1)$. Since f is an n variable function then $Pr(f = 1) = wt(f)/2^n$, but f_j has $n - m$ variables and so $Pr(f_j = 1) = wt(f_j)/2^{n-m}$. By equating these probabilities, we find that $wt(f_j) = wt(f)/2^m$. □ In [42], Xiao and Massey provided a characterization of correlation immunity based on the Walsh transform.

Theorem 6. *[42] A Boolean function on F_2^n is correlation immune of order m iff $W_f(a) = 0$ for all $a \in F_2^n$ with $1 \leq wt(a) \leq m$.*

The original proof of Theorem 6 is very complicated but there is an easier one provided by Sarkar [35]. A balanced mth order correlation immune Boolean function is called an *m-resilient* Boolean function. Using Theorem 5, we see that a Boolean function is *m-resilient* if $wt(f_j) = wt(f)/2^m = 2^{n-1}/2^m = 2^{n-m-1}$ where f_j is as defined in Theorem 5. Using Theorem 6, we see that a Boolean function is *m-resilient* iff $W_f(a) = 0$ for all $a \in F_2^n$ with $0 \leq wt(a) \leq m$. Any balanced function is considered as 0-*resilient*, while nonbalanced functions are considered as (-1)-*resilient*.
Because of their balancedness, resilient functions are preferred over unbalanced correlation immune functions for cryptographic applications. Let $f(x)$ be the Boolean function in the combining generator depicted in figure 2.2. If $f(x)$ is 1-*resilient*, then the Siegenthaler correlation attack [40] will not be possible on a single LFSR. However, as $f(x)$ is not 2-*resilient*, it is possible to attack a pair of LFSRs which is a more difficult task compared to attacking a single LFSR.

3.3 Criteria related to the periodic autocorrelation function

Differential cryptanalysis [2] analyses the effect of particular differences in input pairs on the differences of the resultant output pairs. That is having x and $x + a$ as an input pair, trying to analyse the difference in their output $f(x) + f(x + a)$, and hoping to discover statistical patterns in the output distribution. Computing $Pr(f(x) = f(x + a))$ gives information about the distribution of $f(x) + f(x + a)$. Following the same steps we used to compute $Pr(f(x) = a \cdot x)$ on page 18, we find that for all $a \in F_2^n$,

$$Pr(f(x) = f(x + a)) = Pr(f(x) + f(x + a) = 0) = \frac{1}{2} + \frac{1}{2^{n+1}}\Delta_f(a)$$
$$Pr(f(x) = f(x + a) + 1) = Pr(f(x) + f(x + a) = 1) = \frac{1}{2} - \frac{1}{2^{n+1}}\Delta_f(a)$$

The above equations show that a Boolean function f has resistance against differential cryptanalysis if for most nonzero a, $|\Delta_f(a)|$ is zero or very close to zero as the Boolean function $f(x) + f(x + a)$ will be balanced or close to balanced which means it has an identical or semi-identical distribution which is unwelcomed by cryptanalysts.

Linear Structure

If the derivative of a Boolean function f, $D_a(f)$, is a constant function, then the vector a is called a linear structure of f. The all-zero vector is a trivial linear structure. So when talking about linear structures we mean the nonzero linear structures. If a is a linear structure of f, then $f(x) = f(x + a) + b$, where $b \in F_2$. Existence of such an equality makes the differential attack possible because the distribution of $f(x) + f(x + a)$ will either be a set of zeros or a set of ones which will lead to statistical analysis. So cryptographic functions used in block ciphers must have no nonzero structure.

Propagation Characteristic and Strict Avalanche Criteria

In block ciphers, the coordinate functions of S-boxes are Boolean functions. Propagation Characteristic(PC) and Strict Avalanche Criteria(SAC) are important properties of Boolean functions that are used in S-boxes. The SAC was introduced by Webster and Tavares [45] when they were looking for principles for designing DES-like encryption algorithms. A Boolean function satisfies the SAC if complementing a single bit results in the output of the function being complemented with a probability of a half. This is exactly having $f(x) + f(x + a)$ being balanced for all $a \in F_2^n$ with $wt(a) = 1$. In terms of periodic autocorrelation, this is having $|\Delta_f(a)| = 0$ for all $a \in F_2^n$ with $wt(a) = 0$. In terms of linear structures, f satisfies SAC, if any $a \in F_2^n$ with $wt(a) = 1$ is a linear structure. But the problem with SAC functions is that they can have a large number of vectors with Hamming weight larger than one as their linear structures. This makes employing SAC functions in S-boxes a potential risk. Therefore the SAC was generalized to PC by Preneel [29]. A Boolean function satisfies the propagation characteristic of degree l, $PC(l)$, if complementing l or less bits results in the output of f being complemented with a probability of a half. This is exactly having $f(x) + f(x + a)$ being balanced for all $a \in F_2^n$ with $wt(a) = l$. In terms of the periodic autocorrelation function, f satisfies $PC(l)$ if $\Delta_f(a) = 0$ for all $a \in F_2^n$ such that $1 \leq wt(a) \leq l$. In terms of linear structures f satisfies $PC(l)$, if any $a \in F_2^n$ with $wt(a) = l$ is a linear structure. Boolean functions particularly in block ciphers must satisfy the PC at a high level to avoid differential cryptanalysis. A stronger property than $PC(l)$ is *PC(l) of order k*, which is satisfied when at most k coordinates of the input x are fixed and f still satisfy $PC(l)$. In other words, $D_a(f)$ is *k-resilient* for all $a \in F_2^n$ such that $1 \leq wt(a) \leq l$.

Global Avalanche Characteristics(GAC)

A Boolean function satisfying $PC(l)$ does not have linear structures with Hamming weight less than l. However, the $PC(l)$ criterion does not prevent the possibility of having linear

structures of Hamming weight more than l. This suggests that even the PC which was generalized from SAC is not a sufficient indicator to identify the possibility of differential attacks. On the other hand, the requirement of $f(x) + f(x + a)$ to be 100% balanced to satisfy PC is very strict which leads to having functions satisfying $PC(n)$, and these are bent functions which have nice properties but may be undesirable in practice due to their unbalancedness.

The above shortcomings of PC were stated by Zhang and Zheng [46] as an answer to why they proposed the GAC indicators. GAC indicators consist of a sum-of-squares indicator, defined by $\sigma_f = \sum_{a \in F_2^n} \Delta_f(a)^2$ and an absolute indicator defined by $\Delta_{max} = \max_{a \in F_2^n, a \neq 0} |\Delta_f(a)|$. The smaller σ_f and Δ_f, the better f will be in resisting differential cryptanalysis.

3.4 Other Criteria

Algebraic degree

Cryptographic functions should have high algebraic degrees. A stream cipher using the filter generator to generate its keystream can be attacked by the powerful $O(D^2)$ Berlekamp-Massey linear complexity attack [23] if we know at least $2D$ keystream bits where $D = \sum_{i=0}^{d} \binom{n}{i}$, n is the length of the LFSR used by the filter generator, and d is the algebraic degree of the filter function. If d is low, then the number of key bits, needed to be known, will be small, thereby reducing the amount of the key needed to be known and also the amount of time needed to perform the attack. In the S-boxes of block ciphers, using Boolean functions of low degrees might make higher differential attacks effective [4]. But Boolean functions of degree $n - 1$ or n are not always good functions as they may be weak with respect to some other criteria such as nonlinearity and resilience as we will see in Chapter 3.

Algebraic Immunity ([4, 32])

In 2003, Courtois and Meier [7] introduced a powerful attack called algebraic attack that changed the map of Boolean functions cryptographic criteria. It recovers the secret key by solving a system of nonlinear equations over the finite field F_2. For instance, an algebraic attack on the filter generator depicted in figure 2.2, finds the initial states $(s_0, s_1, ..., s_{n-1})$ provided that we know some part of the keystream. This initial state will then generate the whole keystream. All the attacks we noted previously on the filter generator operate in this way but the advantage of the algebraic attack over those attacks is that it requires a lesser amount of keystream to be known in advance. To better understand algebraic immunity, we need to explain how algebraic attacks work. Let $y_0, y_1, ..., y_k$ be the known part of the keystream. We see that,

$$y_j = f(L_j(s_0, ..., s_{n-1})) \qquad for \quad 0 \leq y_j \leq k-1 \tag{3.11}$$

where L_j is a linear function that updates the LFSR from state j to state $j+1$ (i.e. $L_j(s_0, \ldots, s_{n-1}) = (s_j, s_{j+2}, \ldots, s_{j+n-1})$ with each element written in terms of $(s_0, s_1, \ldots, s_{n-1})$. Now we have a nonlinear system of k equations. Applying linearization to this system we end up with a linear system of k equations and a number of variables $\leq$ to $D = \sum_{i=1}^{d} \binom{n}{i}$ which can be solved by Gaussian elimination in $O(D^3)$ linear operations. But if D is too high then Gaussian elimination will be infeasible. Courtois and Meier came up with an idea that dramatically decreases D. They introduced the notion of an *annihilator* of a Boolean function. They defined it as follows, A nonzero Boolean function g is called an *annihilator* of a Boolean function h if $hg = 0$. Their idea relies on finding an *annihilator* of f and $f+1$ where f is the filter generator function. They proved the existence of an annihilator for both f and $f+1$. They also proved that the degree of an annihilator of a Boolean function on F_2^n is $\leq \lceil \frac{n}{2} \rceil$. To see how this decreases D, suppose that g_0 and g_1 are annihilators for f and $f+1$ respectively. Let $S_0 = (s_0, \ldots, s_{n-1})$. Now when $y_j = 0$, we multiply each equation in 3.11 by g_1, we find that $0 = g_1(L_j(S_0))f(L_j(S_0)) = g_1(L_j(S_0))(f(L_j(S_0)) + 1 + 1) = g_1(L_j(S_0))(f(L_j(S_0)) + 1) + g_1(L_j(S_0)) = g_1(L_j(S_0))$. When $y_j = 1$, we multiply each equa-

tion by g_0, and find that $g_0(L_j(S_0)) = g_0(L_j(S_0))f(L_j(S_0)) = 0$. So the system of equations in 3.11 will be altered to

$$0 = \begin{cases} g_1(L_j(S_0)) & \text{if } y_j = 0 \\ g_0(L_j(S_0)) & \text{if } y_j = 1 \end{cases}$$

Applying linearization to the above system of equations reduces the number of variables to at most $D_1 = \sum_{i=1}^{d_1} \binom{n}{i}$ where $d_1 = min(d(g_0), d(g_1)) \leq \lceil \frac{n}{2} \rceil$. The smaller d_1, the lesser amount of time is needed for solving the above system by Gaussian elimination since D_1 will be much smaller than D. This attack gives rise to the notion of *algebraic immunity*. The *algebraic immunity* of a Boolean function f, denoted by $AI(f)$, is defined as the smallest degree of any nonzero Boolean function g such that $gf = 0$ or $g(f+1) = 0$. We mentioned previously that Courtois and Meier proved that the degree of any annihilator is $\leq \lceil \frac{n}{2} \rceil$. This obviously means that $AI(f) \leq \lceil \frac{n}{2} \rceil$. So to be safe against algebraic attacks, we need to have $AI(f)$ as high as possible.

Bentness

Bent functions on F_2^n are Boolean functions whose nonlinearity achieve the universal nonlinearity bound $2^{n-1} - 2^{\frac{n}{2}-1}$. They have a flat Walsh spectrum, $|W_f(a)| = 2^{\frac{n}{2}}$ for all $a \in F_2^n$. The fraction $\frac{n}{2}$ proves that bent functions exist only when n is even since the Walsh transform is an integer valued function; They can also be defined as the Boolean functions with $PC(n)$, i.e., $\Delta_f(a) = 0$ for all $a \neq 0$, $a \in F_2^n$, and this is because f is bent if and only if $f(x) + f(x+a)$ is balanced for all $a \neq 0$ [4]. Although these two definitions show that bent functions have maximum possible nonlinearity and propagation characteristic, which are nice properties, for a cryptographic function, bent functions are not always ideal due to their unbalancedness which can be proved by observing that balanced functions satisfy $W_f(0) = 0$ which contradicts the bentness property of having the absolute Walsh transform equals to $2^{\frac{n}{2}}$ for all $a \in F_2^n$.

Chapter 4

Theoretical Bounds on Boolean functions criteria

Some of the most important criteria a cryptographic function must satisfy in order to be currently used in practice are balancedness, high algebraic degree, high nonlinearity, high correlation immunity, high propagation characteristic, low absolute indicator, low sum-of-squares indicator and high algebraic immunity. In this chapter, we study the theoretical bounds on these criteria and the best trade-off possible among some of these criteria.

4.1 Bounds on algebraic degree

Siegenthaler's bound [39] states that *an n variable, mth correlation immune Boolean function f has degree at most $n-m$. Moreover, if f is balanced and $m < n-1$, then the degree of f is at most $n-m-1$.* Siegenthaler's proof to the above bound is very complicated but there is an easier proof by Sarkar in [35]. This bound suggests that there is a trade-off between the algebraic degree and correlation immunity, that is, if a cryptosystem has high algebraic degree to resist the linear complexity attack then it will have a low correlation immunity and hence be weak against correlation attacks. So we have to make the best trade-off between algebraic degree and correlation immunity by selecting Boolean functions achieving Siegenthaler's bound, that is, n-variable, m-resilient Boolean functions with degree $d = n-m-1$. Another algebraic degree bound is on bent functions. It was found by Rothaus [30] and it states that *The algebraic degree of any bent function on F_2^n is $\lceil \frac{n}{2} \rceil$ where n is an even integer ≥ 4.*

4.2 Bounds on Nonlinearity

The universal nonlinearity bound $nl(f) \leq 2^{n-1} - 2^{\frac{n}{2}-1}$ is achieved only by bent functions which exist only for even n. Obviously, for odd n the universal bound cannot be tight but the maximum nonlinearity for odd n lies between $2^{n-1} - 2^{\frac{n-1}{2}}$ (It is called the bent concatenation bound and it can be achieved by quadratic functions [4]) and $2^{n-1} - 2^{\frac{n}{2}-1}$. It has been shown that for $n = 1, 3, 5$ and 7, it equals $2^{n-1} - 2^{\frac{n-1}{2}}$ [20]. In 1983, it has been shown that for odd $n \geq 15$, Boolean functions with nonlinearity $\geq 2^{n-1} - 2^{\frac{n-1}{2}}$ can be constructed. In 2006, it has been shown that for $n = 9$, Boolean functions with nonlinearity equal to $2^{n-1} - 2^{\frac{n-1}{2}} + 2^{\frac{n-9}{2}}$ could be constructed [20]. In 2007, it has been shown that for $n = 9$ [18], Boolean functions with nonlinearity equal to $2^{n-1} - 2^{\frac{n-1}{2}} + 2.2^{\frac{n-9}{2}}$ could be constructed. In 2007, a balanced 13-variable Boolean function with nonlinearity equal to $2^{n-1} - 2^{\frac{n-1}{2}} + 2.2^{\frac{n-13}{2}}$ was constructed by Maitra [20].

Bounds on nonlinearity of Balanced functions

Balanced Boolean functions cannot achieve the universal nonlinearity bound but they have a slightly improved upper bound [37],

$$nl(f) \leq \begin{cases} 2^{n-1} - 2^{\frac{n}{2}-1} - 2 & \text{if } n \quad is \quad even \\ \lfloor \lfloor 2^{n-1} - 2^{\frac{n}{2}-1} \rfloor \rfloor & \text{if } n \quad is \quad odd \end{cases} \tag{4.1}$$

where $\lfloor \lfloor x \rfloor \rfloor$ denotes the maximum even integer less than or equal to x. $nl(f) = \min_{a \in F_2^n} d(f, a \cdot x) = \min_{a \in F_2^n} wt(f \oplus a \cdot x) = wt(f) + wt(a \cdot x) - 2wt(f \cdot (a \cdot x))$ but $wt(f)$ and $wt(a \cdot x)$ are even since they are balanced functions, so $nl(f)$ is an even integer. So the maximum nonlinearity possible for balanced functions can be upper bounded by the largest even number less than $2^{n-1} - 2^{\frac{n}{2}-1}$ which is $2^{n-1} - 2^{\frac{n}{2}-1} - 2$ for even n and $\lfloor \lfloor 2^{n-1} - 2^{\frac{n}{2}-1} \rfloor \rfloor$ for odd n. This proves the above upper bounds. The maximum nonlinearity of balanced functions is unknown for any $n \geq 8$. For instance, the highest nonlinearity currently known for an 8-variable balanced function is 116 which is less than 118 (the number obtained by the

above upper bound) but still there is no proof that there is no balanced Boolean function with nonlinearity 118. For odd n it is known that the above upper bound is not tight since for $n = 7$, the maximum nonlinearity of balanced functions is known to be 56 which is less than $\lfloor\lfloor 2^{7-1} - 2^{\frac{7}{2}-1}\rfloor\rfloor = \lfloor\lfloor 58.34\rfloor\rfloor = 58$ (the number obtained by the above upper bound). For even n, it is still unknown whether the above upper bound is tight or not as all the currently known maximum nonlinearities ($n \leq 6$) of balanced functions equal the above upper bound which do not affirm or negate the tightness of the above upper bound.

Bounds on nonlinearity of m-resilient functions

A lot of research has been done on optimizing the nonlinearity while fixing the order of resilience. In 1999, Pasalic and Johansson [28] answered the question of whether an n-variable Boolean function, $n \leq 6$, which is m-resilient and with nonlinearity $nl(f)$, exists or not by solving the following integer programming(0-1 IP) model:

$$\hat{f}(0) = 2^{n-1}, \quad \textit{balancedness}$$

$$\hat{f}(a) = 0, \quad 1 \leq wt(a) \leq m, \quad \textit{m-resilient},$$

$$\left|\hat{f}(a)\right| \leq 2^{n-1} - nl(f), \quad a \neq 0, \quad \textit{nonlinearity}$$

The above 0-1 IP has no objective function as any solution is sufficient to solve the problem. If the above 0-1 IP model has no solution for a specified resilience(m) and nonlinearity($nl(f)$) then there does not exist a Boolean function having that specified resilience and nonlinearity. Since IP is an NP-hard problem and problems of more than 100 variables are considered as infeasible, the above model can only be used for $n \leq 6$ as for $n = 7$ we have 128 variables which is infeasible. Pasalic and Johansson solved the above model for $n \leq 6$ but their main result was the following theorem found when $n = 6$.

Theorem 7. *[28] The maximum nonlinearity for a 1-resilient function on 6 variables is 24.*

Although many useful results between resilience and nonlinearity were found by the approach of fixing the order of resilience and optimizing nonlinearity, this approach has not investigated the exact nature of the trade-off between resilience and nonlinearity, that is, it has not answered the question, what is the maximum nonlinearity of an n-variable and m-resilient Boolean function? In 2000, this question was answered independently by Sarkar and Maitra [33], by Tarannikov [44] and by Zhang and Zheng [48].

Sarkar and Maitra found the following major result on the Walsh Spectrum of resilient or correlation immune functions which consequently allowed them to obtain a nontrivial upper bound on the maximum possible nonlinearity of m-resilient or m-correlation immune functions.

Theorem 8. *[33] Let f be an n-variable with $n \geq 3$. Then for all $a \in F_2^n$,*

$$W_f(a) \equiv \begin{cases} 0 \ (mod\, 2^{m+1}) & \text{if } f \text{ is } m\text{-correlation immune, } m \leq n-2 \\ 0 \ (mod\, 2^{m+2}) & \text{if } f \text{ is } m\text{-resilient, } m \leq n-3 \end{cases}$$

For the proof of this theorem, refer to [33]. Theorem 8 was used to deduce the following theorem,

Theorem 9. *[33] Let $nlc(n,m)$ denote the maximum nonlinearity of an n-variable and m-correlation immune Boolean function, $nlr(n,m)$ denote the maximum nonlinearity of an n-variable and m-resilient Boolean function and $nlmax(n)$ denote the maximum nonlinearity for an n-variable Boolean function. If n is even, then*

$$nlc(n,m) \leq \begin{cases} 2^{n-1} - 2^m & \text{if } m > \frac{n}{2} - 1 \\ 2^{n-1} - 2^{\frac{n}{2}-1} - 2^m & \text{if } m \leq \frac{n}{2} - 1 (*) \end{cases}$$

$$nlr(n,m) \leq \begin{cases} 2^{n-1} - 2^{m+1} & \text{if } m+1 > \frac{n}{2} - 1 \\ 2^{n-1} - 2^{\frac{n}{2}-1} - 2^{m+1} & \text{if } m+1 \leq \frac{n}{2} - 1 (*) \end{cases}$$

If n is odd, then

$$nlc(n,m) \leq \begin{cases} 2^{n-1} - 2^m & \text{if } nlmax(n) > 2^{n-1} - 2^m \\ \max_{h\geq 0}\{h2^m\} \leq nlmax(n) & \text{if } nlmax(n) \leq 2^{n-1} - 2^m \ (*) \end{cases}$$

$$nlr(n,m) \leq \begin{cases} 2^{n-1} - 2^{m+1} & \text{if } nlmax(n) > 2^{n-1} - 2^{m+1} \\ \max_{h\geq 0}\{h2^{m+1}\} \leq nlmax(n) & \text{if } nlmax(n) \leq 2^{n-1} - 2^{m+1}(*) \end{cases}$$

Further in all $nlr(m,n)$ *bounds except the starred ones, the Walsh spectrum of any function achieving the stated bounds has three values* $0, \pm 2^{m+2}$*. Also in all* $nlc(m,n)$ *bounds except the starred ones, the Walsh spectrum of any function achieving the stated bounds has three values* $0, \pm 2^{m+1}$*.*

Proof:[33]

We prove the cases concerning $nlr(n,m)$. The other cases concerning $nlc(n,m)$ being similar. We know that for all $a \in F_2^n$, $W_f(a) = 2^n - d(f, a\cdot x)$. Using Theorem 8 we replace $W_f(a)$ by $\pm k2^{m+2}$ where $k \geq 0$, thus we find that $d(f, a\cdot x) = 2^{n-1} \pm k2^{m+1}$. Now we have 4 cases,

Case 1: If n is even and $m+1 > \frac{n}{2} - 1$; $nlr(n,m) = \min_{a\in F_2^n} d(f, a\cdot x) = \min_{k\geq 0}\{2^{n-1} \pm k2^{m+1}\} \leq 2^{n-1} - 2^{m+1}$(as k cannot be 0 for all a). Since $2^{n-1} - 2^{m+1} < nlmax(n) = 2^{n-1} - 2^{\frac{n}{2}-1}$. Therefore, $nlr(n,m) \leq 2^{n-1} - 2^{m+1}$ is a nontrivial upper bound.

Case 2: If n is even and $m+1 \leq \frac{n}{2} - 1$; Let $2^{\frac{n}{2}-1} = p2^{m+1}$(as $m+1 \leq \frac{n}{2} - 1$), $nlr(n,m) = \min_{a\in F_2^n} d(f, a\cdot x) = \min_{k\geq 0}\{2^{n-1} \pm k2^{m+1}\}$. If for all a we have $k \leq p$, then f must be bent(as $nlr(n,m) = 2^{n-1} - 2^{\frac{n}{2}-1}$) and hence not resilient. Thus there must be some a such that the corresponding $k > p$. This shows that $nlr(n,m) = 2^{n-1} - 2^{\frac{n}{2}-1} - 2^{m+1}$.

Case 3: If n is odd and $nlmax(n) > 2^{n-1} - 2^{m+1}$; this case is similar to *Case 1.*

Case 4: If n is odd and $nlmax(n) \leq 2^{n-1} - 2^{m+1}$; $nlr(n,m) = \min_{a\in F_2^n} d(f, a\cdot x) = \min_{k\geq 0}\{2^{n-1} \pm k2^{m+1}\} = \min_{k\geq 0}\{2^{n-1} - k2^{m+1}\}$. It is obvious that $nlr(n,m)$ is a multiple of 2^{m+1}. Therefore the highest multiple of 2^{m+1} which is less than or equal to $nlmax(n)$ is a nontrivial upper bound for $nlr(n,m)$.

We prove the last statement for the $nlr(n,m)$ case, the other case is similar. If the Walsh

spectrum for any function achieving the bounds does not have three value, $0, \pm 2^{m+2}$, then from the divisibility property we have $W_f(a) = \pm 2^{m+i}$ for some a and for some $i \geq 3$. Thus $d(f, a \cdot x) = 2^n - W_f(a) = 2^n - 2^{m+i}$ which is less than the stated bounds.□

In [48], Zheng and Zhang improved the upper bound of correlation immune functions in Theorem 9. They showed that the nonlinearity of an mth order correlation immune and n-variable Boolean function f is less than or equal to $2^{n-1} - 2^{m+1}$ if $m \geq 0.6n - 0.4$, regardless of the balancedness of f. Boolean functions whose Walsh spectrum has three values $0, \pm 2^{\lambda}$ where λ is a positive integer are called plateaued functions. So Boolean functions achieving the above stated bounds are called plateaued functions. The following theorem proved by Claude Carlet [4] suggested a bound for $nlr(n, m)$ regardless of the evenness of n.

Theorem 10. *[4] Let f be an n-variable and m-resilient Boolean function then, $nlr(n, m) \leq 2^{n-1} - 2^{m+1} \lceil \frac{2^{n-m-2}}{\sqrt{2^n - \sum_{i=0}^{m} \binom{n}{i}}} \rceil$*

Proof:[4]

From Sarkar-Maitra's divisibility property, we know that $W_f(a) = \varphi(a) 2^{m+2}$ where $\varphi(a)$ is an integer valued function. But Parseval's equation (see Theorem 4) and the fact that $W_f(a) = 0$ for all a with weight $\leq m$ imply that

$$\sum_{a/wt(a)>m} \varphi(a)^2 = 2^{2n-2m-4}$$

and thus

$$\max_{a \in F_2^n} |\varphi(a)| \geq \sqrt{\frac{2^{2n-2m-4}}{2^n - \sum_{i=0}^{m} \binom{n}{i}}} = \frac{2^{n-m-2}}{\sqrt{2^n - \sum_{i=0}^{m} \binom{n}{i}}}$$

Since $\varphi(a)$ is an integer valued function, then $\max_{a \in F_2^n} |\varphi(a)| \geq \lceil \frac{2^{n-m-2}}{\sqrt{2^n - \sum_{i=0}^{m} \binom{n}{i}}} \rceil$. This implies that,

$nlr(n, m) = 2^{n-1} - \frac{1}{2} \max_{a \in F_2^n} |W_f(a)| = 2^{n-1} - 2^{m+1} \max_{a \in F_2^n} |\varphi(a)| \leq 2^{n-1} - 2^{m+1} \lceil \frac{2^{n-m-2}}{\sqrt{2^n - \sum_{i=0}^{m} \binom{n}{i}}} \rceil$.

□

Applying the same approach of Theorem 10 but for m-correlation immune functions, we find a bound similar to that of Theorem 10.

$$nlc(n, m) \leq 2^{n-1} - 2^m \lceil \frac{2^{n-m-1}}{\sqrt{2^n - \sum_{i=1}^{m} \binom{n}{i}}} \rceil \tag{4.2}$$

Claude Carlet [4] suggested that his upper bound in Theorem 10 is potentially better than Sarkar-Maitra's bound when n is even but when n is odd he pointed that it is difficult to say which is better as Sarkar-Maitra's bound involves $nlmax(n)$ which is unknown for $n \geq 9$ which in turn makes Theorem 10's bound more effective than Sarkar-Maitra's bound.

Bounds on nonlinearity of m-resilient functions with degree d

The following theorem introduced by Claude Carlet [5] generalized Sarkar-Maitra's divisibility property by involving the algebraic degree d. We will call it the degree divisibility property.

Theorem 11. *[5] Let f be an n-variable m-resilient function and let d be its algebraic degree. The values of the Walsh spectrum of f are divisible by $2^{m+2+\lfloor \frac{n-m-2}{d} \rfloor}$.*

For the proof of Theorem 11, refer to [34]. It was first proved by using the numerical normal form[1] but later it was proved by Sarkar [34] using only the properties of the Walsh spectrum. Sarkar also proved that the values of the Walsh spectrum of any m-correlation immune function are divisible by $2^{m+1+\lfloor \frac{n-m-1}{d} \rfloor}$ [34]. Theorem 11 gives directly a more precise upper bound on the nonlinearity of an m-resilient function of degree d.

Theorem 12. *[5] Let f be n-variable m-resilient Boolean function with degree d, then if $\frac{n}{2} - 1 < m+1+\lfloor \frac{n-m-2}{d} \rfloor$ we have the maximum nonlinearity $nlr(n,m,d) \leq 2^{n-1} - 2^{m+1+\lfloor \frac{n-m-2}{d} \rfloor}$; otherwise we have*

$$nlr(n,m,d) \leq \begin{cases} 2^{n-1} - 2^{\frac{n}{2}-1} - 2^{m+1+\lfloor \frac{n-m-2}{d} \rfloor} & \text{if } n \text{ is even} \\ 2^{n-1} - 2^{m+1+\lfloor \frac{n-m-2}{d} \rfloor} \lceil 2^{\frac{n}{2}-m-2-\lfloor \frac{n-m-2}{d} \rfloor} \rceil & \text{if } n \text{ is odd} \end{cases}$$

For the proof of Theorem 12, refer to [5]. An upper bound similar to that of Theorem 12 but for m-correlation immune functions can be given by using the degree divisibility property for correlation immune functions (It can be obtained by replacing $2^{m+1+\lfloor \frac{n-m-2}{d} \rfloor}$

[1] The Numerical Normal Form (NNF) is a representation of Boolean functions introduced by Carlet-Guillot[6].

with $2^{m+\lfloor\frac{n-m-1}{d}\rfloor}$ in each bound in Theorem 12). Carlet also showed in [5], that the nonlinearity of an m-correlation immune and n-variable Boolean function is less than or equal to $2^{n-1}-2^{m+1+\lfloor\frac{n-m-2}{d}\rfloor}$ if $m \geq 0.6n$. The m-resilient functions achieving the bound $2^{n-1}-2^{m+1}$ when $m+1 > \frac{n}{2}-1$ also achieve Siegenthaler's degree bound $n-m-1$ which can be proved as follows: From Theorem 12, we see that m-resilient functions achieve the bound $2^{n-1}-2^{m+1}$ when $d > n-m-2$, but $d \leq n-m-1$, thus $d = n-m-1$.

Bounds on nonlinearity involving the GAC indicators

In the previous Chapter we stated that the GAC indicators are the absolute indicator(Δ_{max}) and the sum-of-squares indicator(σ_f). The following theorem [47] gives a straightforward upper bound on nonlinearity involving the sum-of-squares indicator(σ_f).

Theorem 13. *[47] For any function f on F_2^n the nonlinearity of f satisfies*

$$nl(f) \leq 2^{n-1} - \tfrac{1}{2}\sqrt[4]{\sigma_f}$$

Proof:[47]

Theorem 2(Wiener-Khintchine) suggests that $\Delta_f(a) = \sum_{u\in {F_2}^n} W_f(u)^2(-1)^{a.u}$, squaring both sides of this equation and taking the sum for all $a \in F_2^n$, we get the following equation $\sum_{a\in F_2^n} W_f(a)^4 = 2^n \sum_{a\in F_2^n} \Delta_f(a)^2$ [47]. Thus there exists an $a \in F_2^n$, such that $W_f(a)^4 \geq \sum_{a\in F_2^n} \Delta_f(a)^2$. This implies that $W_f(a) \geq \sqrt[4]{\sum_{a\in F_2^n} \Delta_f(a)^2} = \sqrt[4]{\sigma_f}$. Hence from equation 3.7, we have

$$nl(f) = 2^{n-1} - \frac{1}{2} max_{a\in F_2^n} |W_f(a)| \leq 2^{n-1} - \frac{1}{2}\sqrt[4]{\sigma_f}. \quad \square$$

The following theorem gives an upper bound on nonlinearity involving the absolute indicator(Δ_{max}).

Theorem 14. *[47] For any function f on F_2^n, the nonlinearity of f satisfies*

$$nl(f) \leq 2^{n-1} - \tfrac{1}{2}\sqrt{2^n + \Delta_{max}}$$

For the proof of this theorem, refer to [47].

4.3 Bounds on GAC indicators

A lower bound for σ_f is obviously $\sigma_f \geq 2^{2n}$ since $\Delta_f(0) = 2^{2n}$. This bound is achieved when the derivative $D_f(a)$ is balanced for all $a \neq 0$. That is when f is bent. An upper bound for σ_f is $\sigma_f \leq 2^{3n}$ since $\Delta_f(a)^2 \leq 2^{2n}$ for all a. Obviously this upper bound is achieved when $D_f(a)$ is constant for all $a \neq 0$. That is when f is an affine function. Therefore, $2^{2n} \leq \sigma_f \leq 2^{3n}$. A lower bound for Δ_{max} is obviously $\Delta_{max} \geq 0$ which as we stated above is achieved when the derivative $D_f(a)$ is balanced for all $a \neq 0$ (f is bent). An upper bound for Δ_{max} is $\Delta_{max} \leq 2^n$ which is achieved when f has a nonzero linear structure. Therefore, $0 \leq \Delta_{max} \leq 2^n$. These are the trivial bounds for the sum-of-squares indicator and the absolute indicator for any Boolean function. But we need to know the GAC bounds for resilient functions since any cryptographic function must be resilient. In [21], Maitra introduced lower bounds for both the sum-of-squares indicator and the absolute indicator on resilient and correlation immune functions. In this section, we discuss all these bounds.

Lower bounds on the sum-of-squares indicator

For any Boolean function f, define F_f as the number of nonzero values on the Walsh spectrum, i.e. $F_f = \#\{a \in F_2^n : W_f(a) \neq 0\}$. A lower bound for the sum-of-squares indicator involving F_f was introduced in [21].

Theorem 15. *[21] Let f be a Boolean function on F_2^n. Then, $\sigma_f \geq \frac{2^{3n}}{F_f}$. Moreover, if f has a three valued Walsh spectrum $0, \pm 2^\lambda$ (f is a plateaued function), then $\sigma_f = \frac{2^{3n}}{F_f}$.*

This theorem was later modified by Maitra to the following theorem.

Theorem 16. *[21] Let f be a Boolean function on F_2^n. Then $\sigma_f \geq \frac{2^{3n}}{2^n - \sum_{i=1}^{m} \binom{n}{i}} > 2^{2n} + 2^{n+log_2 \sum_{i=1}^{m} \binom{n}{i}}$ if f is m-correlation immune and $\sigma_f \geq \frac{2^{3n}}{2^n - \sum_{i=0}^{m} \binom{n}{i}} > 2^{2n} + 2^{n+log_2 \sum_{i=0}^{m} \binom{n}{i}}$ if f is m-resilient.*

Proof:[21]

It is obvious to see that $F_f \leq 2^n - \sum_{i=1}^{m} \binom{n}{i}$ when f is m-correlation immune and $F_f \leq 2^n - \sum_{i=0}^{m} \binom{n}{i}$ when f is m-resilient. Substituting these values of F_f in Theorem 15 yields $\sigma_f \geq \frac{2^{3n}}{2^n - \sum_{i=1}^{m} \binom{n}{i}} > 2^{2n} + 2^{n+log_2 \sum_{i=1}^{m} \binom{n}{i}}$ for m-correlation immune functions and $\sigma_f \geq \frac{2^{3n}}{2^n - \sum_{i=0}^{m} \binom{n}{i}} > 2^{2n} + 2^{n+log_2 \sum_{i=0}^{m} \binom{n}{i}}$ for m-resilient functions. □

Another lower bound on σ_f was deduced by using the divisibility property and the degree divisibility property of resilient and correlation immune Boolean functions described in the previous section. The following theorem shows the sum-of-squares bounds on m-resilient and m-correlation immune functions using the divisibility property,

Theorem 17. *[21] Let f be a Boolean function on F_2^n. Then $\sigma_f \geq 2^{n+2m+2}$ if f is m-correlation immune and $\sigma_f \geq 2^{n+2m+4}$ if f is m-resilient. Moreover, these bounds are achieved when f is a plateaued function.*

Proof:[21]

If f is m-correlation immune; The divisibility property $W_f(a) \equiv 0 (mod\, 2^{m+1})$ tell us that the Walsh values of f are $0, \pm i2^{m+1}$, where i is a positive integer. Using Parseval's equation $\sum_{a \in F_2^n} W_f(a)^2 = 2^{2n}$, we deduce that $F_f \leq 2^{2n-2m-2}$. Substituting this value of F_f in Theorem 15 yields $\sigma_f \geq 2^{n+2m+2}$. If f is plateaued then the Walsh values of f will be $0, \pm 2^{m+1}$ and hence $F_f = 2^{2n-2m-2}$ and so σ_f achieves the bound $\sigma_f = 2^{n+2m+2}$.

Similarly, if f is m-resilient; The divisibility property $W_f(a) \equiv 0 (mod\, 2^{m+2})$ tell us that the Walsh values of f are $0, \pm i2^{m+2}$, where i is a positive integer. Using Parseval's equation $\sum_{a \in F_2^n} W_f(a)^2 = 2^{2n}$, we deduce that $F_f \leq 2^{2n-2m-2}$. Substituting this value of F_f in Theorem 15 yields $\sigma_f \geq 2^{n+2m+4}$. If f is plateaued then the Walsh values of f will be $0, \pm 2^{m+2}$ and hence $F_f = 2^{2n-2m-2}$ and so σ_f achieves the bound $\sigma_f = 2^{n+2m+4}$. □

Since the trivial bound for σ_f is 2^{2n}, then the bound when f is m-correlation immune in Theorem 17 is nontrivial when $n + 2m + 2 > 2n$, that is when $m > \frac{n}{2} - 1$. Similarly, the bound when f is m-resilient in Theorem 17 is nontrivial when $n + 2m + 4 > 2n$, that is when $m > \frac{n}{2} - 2$. The following theorem shows another lower bound for the sum-of-squares

bounds on m-resilient and m-correlation immune functions using the degree divisibility property,

Theorem 18. *[21] Let f be a Boolean function on F_2^n of degree d. Then $\sigma_f \geq 2^{n+2m+2+2\lfloor \frac{n-m-1}{d} \rfloor}$ if f is m-correlation immune and $\sigma_f \geq 2^{n+2m+4+2\lfloor \frac{n-m-2}{d} \rfloor}$ if f is m-resilient. Moreover, these bounds are achieved when f is a plateaued function.*

The proof of this theorem is very similar to the proof of the above theorem. In the previous section we showed that nonlinearity and algebraic degree of resilient or correlation immune functions are optimized simultaneously if f is plateaued. Theorems 16 and 17 tell us also that the sum-of-squares indicator for resilient functions is optimized when f is plateaued. This indicates that for an n-variable, m-resilient plateaued function the nonlinearity, the algebraic degree and the sum-of-squares indicator are optimized simultaneously.

Lower bounds on the absolute indicator

Every lower bound for the sum-of-squares indicator of the form $\sigma_f \geq \sigma$ directly implies that the absolute indicator is lower bounded by $\sqrt{\frac{\sigma - 2^{2n}}{2^n - 1}}$. This is proved as follows [21]: we know that $\sigma_f = \sum_{a \in F_2^n} \Delta_f(a)^2$. Thus the absolute value of each $\Delta_f(a)$ will be the minimum value when each $\Delta_f(a) = 0$, for all $a \in F_2^n$, $a \neq 0$, possess equal values. Hence, the minimum value of Δ_{max} will be $\sqrt{\frac{\sigma_f - 2^{2n}}{2^n - 1}}$. Thus, theorem 16 implies that $\Delta_{max} \geq \sqrt{\frac{1}{2^n - 1} \frac{2^{2n} \sum_{i=1}^{m} \binom{n}{i}}{2^n - \sum_{i=1}^{m} \binom{n}{i}}} > 2^{\frac{n}{2}} \sqrt{\frac{\sum_{i=1}^{m} \binom{n}{i}}{2^n - \sum_{i=1}^{m} \binom{n}{i}}}$ is a lower bound for m-correlation immune functions and $\Delta_{max} \geq \sqrt{\frac{1}{2^n - 1} \frac{2^{2n} \sum_{i=0}^{m} \binom{n}{i}}{2^n - \sum_{i=0}^{m} \binom{n}{i}}} > 2^{\frac{n}{2}} \sqrt{\frac{\sum_{i=0}^{m} \binom{n}{i}}{2^n - \sum_{i=0}^{m} \binom{n}{i}}}$ for m-resilient functions. Similarly other lower bounds can be obtained from Theorems 17 and 18.

Chapter 5

Boolean function database website

The Selmer center is responsible for developing and hosting a Boolean function database website for Europe-wide cryptography project called ECRYPT. The idea is that the website will be used throughout the world, and that researchers into Boolean functions will submit their best functions to the database.

One aim of this database is to show how close the bounds on the properties of Boolean functions in the database come to the theoretical bounds on the properties of Boolean functions. This then motivates the finding of construction methods that achieve the theoretical bounds and also the finding of tighter theoretical bounds that match the properties of the best representative functions in the database. If the bounds on the database equal the theoretical bounds, our database will be completed with respect to the criteria considered, since it will contain a representative for every good function. The following example shows how the database could be used. Suppose you are looking for a Boolean function with certain properties. Then you can consult the database and discover that:

1. There is a function with the properties you want that was found and saved in the database by someone else.

2. There is no function with the properties you want in the database. If your properties are within the theoretical bounds (such theoretical bounds may also be provided by the database), a function with your properties could exist but it is not known how to construct it. This will motivate researchers to either improve the theoretical bounds and/or find construction methods that achieve the current theoretical bounds. If

your properties are not within known theoretical bounds, then a function with your properties does not exist.

We implemented the following three pages in the database website.

Conversion page

In this page, the user can convert between three different Boolean functions representations. These three representations are the truth table representation, the ANF representation, and the trace representation.

Check page

In this page, the user can enter a Boolean function in either its truth table, ANF or trace representation to calculate the following cryptographic criteria: ANF degree, balancedness, bentness, absolute distribution of the coefficients of the Walsh spectrum, nonlinearity, correlation immunity, absolute distribution of the coefficients of the autocorrelation spectrum, propagation characteristic, absolute indicator and sum-of-squares indicator. The user can save the function in the database if the entered function is interesting. The function will be saved in three different formats (TT, ANF, Trace) together with the calculated (mentioned above) criteria.

Search page

In this page, the user can search in the database for a Boolean function by any combination of the following attributes: number of variables (n), ANF degree (d), balancedness criterion (bal), bentness criterion (bent), nonlinearity criterion (nl), correlation immunity criterion (m), propagation characteristic of degree 0 (pc), absolute indicator (abs) and sum-of-squares indicator (sos).The page allows the user to see the lower/upper bounds of the above mentioned attributes for any Boolean function in the database possessing any valid combination of the above mentioned attributes. It also allows the user to see the theoretical

lower/upper bounds for any possible Boolean function possessing any valid combination of the above mentioned attributes. For instance, if the user looks for an 8-variable balanced Boolean function, then the database upper bound on nonlinearity will be 116 (116 is the highest nonlinearity currently known for an 8-variable balanced function) and the theoretical upper bound for nonlinearity will be 118 (See the upper bound on balanced Boolean functions on page 29). We implemented the following theoretical bounds depending on the user input,

Input Combination = n

In this case, the only nontrivial theoretical bound is the universal upper bound on nonlinearity when n is even, if n is odd the upper bound is obtained by taking the floor of the universal upper bound on nonlinearity.

Input Combination = $n + d$

Besides the universal upper bound for nonlinearity, we have Siegenthaler's bound on correlation immunity, $m < n - d$.

Input Combination = $n + bal$

In this case, the only nontrivial bound is the bound in 4.1.

Input Combination = $n + d + bal$

Besides the universal upper bound for nonlinearity, we have Siegenthaler's bound on correlation immunity (in this case resilience as we have 'bal' in the input), $m < n - d - 1$.

Input Combination = $n + bent$

Besides the universal upper bound for nonlinearity, we have Rothaus's bound on the degree of bent functions, $d <= n/2$.

Input Combination $= n + m$

In this case, we have two nontrivial upper bounds: one on nonlinearity and the other on the ANF degree. When n is even, we use the upper bound on nonlinearity in Theorem 9 but when n is odd we use the upper bound on nonlinearity in 4.2 since it is more usable that of Theorem 9 when n is odd. The upper bound on the ANF degree is Siegenthaler's bound $d < n - m$ (if $m \geq 0.6n - 0.4$, we use the upper bound $2^{n-1} - 2^{m+1}$). We also have a lower bound on the sum-of-squares indicator and a lower bound on the absolute indicator. We use the lower bound on the sum-of-squares indicator, σ_f, in Theorem 17 (for m-correlation immune). We use the lower bound on the absolute indicator obtained by substituting the lower bound of σ_f in $\sqrt{\frac{\sigma_f - 2^{2n}}{2^n - 1}}$.

Input Combination $= n + bal + m$

In this case, we have two nontrivial upper bounds: one on nonlinearity and the other on the ANF degree. When n is even, we use the upper bound on nonlinearity in Theorem 9 but when n is odd we use the upper bound on nonlinearity in Theorem 10 since it is more usable than that of Theorem 9 when n is odd. The upper bound on the ANF degree is Siegenthaler's bound $d < n - m - 1$. We also have a lower bound on the sum-of-squares indicator and a lower bound on the absolute indicator. We use the lower bounds on the sum-of-squares indicator, σ_f, in Theorem 17 (for m-resilient). We use the lower bound on the absolute indicator obtained by substituting the lower bound of σ_f in $\sqrt{\frac{\sigma_f - 2^{2n}}{2^n - 1}}$.

Input Combination $= n + d + m$

In this case, we have only one nontrivial upper bound on nonlinearity. It is similar to the one in Theorem 12 and is obtained by simply replacing $2^{m+1+\lfloor \frac{n-m-2}{d} \rfloor}$ by $2^{m+\lfloor \frac{n-m-1}{d} \rfloor}$ in each bound in Theorem 12 (if $m \geq 0.6n$, we use the upper bound $2^{n-1} - 2^{m+1+\lfloor \frac{n-m-2}{d} \rfloor}$). We have two nontrivial lower bounds, one lower bound on the sum-of-squares indicator and the other lower bound on the absolute indicator. We use the lower bound on the sum-of-squares

indicator, σ_f, in Theorem 18 (for m-correlation immune). We use the lower bound on the absolute indicator obtained by substituting the lower bound of σ_f in $\sqrt{\frac{\sigma_f - 2^{2n}}{2^n - 1}}$.

Input Combination $= n + d + bal + m$

In this case, we have only one nontrivial upper bound on nonlinearity. It is the one in Theorem 12. We have two nontrivial lower bounds, one on the sum-of-squares indicator and the other lower bound on the absolute indicator. We use the lower bound on the sum-of-squares indicator, σ_f, in Theorem 18 (for m-resilient). We use the lower bound on the absolute indicator obtained by substituting the lower bound of σ_f in $\sqrt{\frac{\sigma_f - 2^{2n}}{2^n - 1}}$.

Input Combination $= n + abs$

In this case, we have two nontrivial upper bounds: one on nonlinearity and the other on correlation immunity. The upper bound on nonlinearity is the one in Theorem 14. The upper bound on correlation immunity is $m \leq \lceil (\log_2 (\Delta^2_{max}(2^n - 1) + 2^{2n}) - n - 2)/2 \rceil$ which can be deduced from Theorem 17.

Input Combination $= n + sos$

In this case, we have two nontrivial upper bounds: one on nonlinearity and the other on correlation immunity. The upper bound on nonlinearity is the one in Theorem 13. The upper bound on correlation immunity is $m \leq \lceil \frac{\log_2 \sigma_f - n - 2}{2} \rceil$ which can be deduced from Theorem 17.

Our contribution to the database website

The Boolean function database was initially created by Lars Erik Danielsen and later developed by Sondre Ronjom. The following are our contributions to their work

1. Implementing the conversions from trace representations to a truth table or ANF representations and vice versa.

2. Enabling the calculations of cryptographic criteria for Boolean functions with number of variables up to 21.

3. Calculating theoretical bounds on the cryptographic criteria implemented in the database.

The following table summarizes the implemented theoretical bounds in the website.

Input/Bounds	d	nl	m	pc	abs	sos
n		U				
$n+d$		U	U			
$n+bal$		U				
$n+d+bal$		U				
$n+bent$	U	L=U	L=U	L=U	L=U	L=U
$n+m$	U	U			L	L
$n+bal+m$	U	U			L	L
$n+d+m$		U			L	L
$n+d+bal+m$		U			L	L
$n+abs$		U	U			
$n+sos$		U	U			

Table 5.1: The rows on the table refer to the user input, while the columns refer to the bound on the implemented criteria. Entry 'U' means that the bound implemented is an upper bound, while entry 'L' means that the bound implemented is a lower bound and the empty entry means that the bound is either trivial or nonexistent.

Chapter 6

Golay complementary sequences and arrays

In the previous chapters, we studied the Walsh spectrum and the autocorrelation spectrum of a Boolean function and some of the cryptographic criteria related to them. The rest of this thesis deals with more spectral measures of a Boolean function, including aperiodic autocorrelation and presents new constructions with respect to these spectral measures, related to the Golay construction for complementary sequences. In this chapter, we give a survey on Golay sequence and array pairs. This survey will serve as an introduction to the next chapter.

6.1 Golay complementary binary sequences

Let $A = (A_0, A_1, \ldots, A_{N-1})$ be a *binary sequence of length* N such that $A_i \in \{1, -1\}$ for $0 \leq i \leq N-1$. The *aperiodic autocorrelation function* of a length s binary sequence A is given by

$$C_A(k) = \sum_{i=0}^{N-k-1} A_i A_{i+k} \tag{6.1}$$

for $0 \leq k \leq N-1$ and measures the extent to which a binary sequence repeats itself.
Let $B = (B_0, B_1, \ldots, B_{N-1})$ be another binary sequence of length N. The pair (A, B) is called a *Golay complementary sequence pair* if

$$C_A(k) + C_B(k) = 0 \text{ for all } k \neq 0.$$

We call a sequence A a *Golay sequence* if it forms a Golay sequence pair with some sequence B. The earliest Golay complementary sequences were defined over a binary alphabet, $\{1, -1\}$. They were introduced by M. Golay in 1949. Golay showed the existence of complementary pairs of lengths 2 and 10 [13], and 26 [15]. He also gave a construction for pairs of length $2N$ and $2MN$ from existing pairs of lengths M and N [14] (see section 6.1.2). In 1974, Turyn gave a construction for pairs of length MN from existing pairs of lengths M and N [41](see section 6.1.2). These constructions tell us that Golay pairs exist for infinitely many lengths. The following two theorems summarize the known results about the existence of Golay sequences.

Theorem 19. *[41](see section 6.1.2) If there exist binary Golay sequence pairs of length N and M then there exists a binary Golay sequence pair of length NM.*

Corollary 1. *There exists a binary Golay sequence pair of length $2^a 10^b 26^c$ for all integer $a, b, c \geq 0$.*

The following two theorems summarize the known results about the nonexistence of Golay sequences.

Theorem 20. *[14] If there exists a binary Golay sequence pair of length $N > 1$ then N is even.*

Theorem 21. *[11] If there exists a binary Golay sequence pair of length $N > 1$ then N has no prime factor congruent to 3 modulo 4.*

6.1.1 Spectral property of Golay binary sequences

A binary Golay sequence A of length N can be associated with a polynomial $A(z) = A_0 + A_1 z + A_2 z^2 + \ldots + A_{N-1} z^{N-1}$ in indeterminate z with coefficients ± 1.

Theorem 22. *[26] A pair (A, B) of length N is called a binary Golay pair if the associated polynomials $(A(z), B(z))$ satisfy*

$$A(z)A(z^{-1}) + B(z)B(z^{-1}) = 2N \tag{6.2}$$

Moreover, if z is restricted to lie on the unit circle in the complex plane, i.e. $|z| = 1$. *Then*

$$|A(z)|^2 + |B(z)|^2 = 2N, \quad |z| = 1 \tag{6.3}$$

Proof:

The Golay property, $C_A(k) + C_B(k) = 0$ for $k \neq 0$, and equation 6.2 are equivalent because $A(z)A(z^{-1}) = C_A(0) + \sum_{k=1}^{N-1} C_A(k)(z^k + z^{-k})$. If $|z| = 1$ then $|A(z)|^2 = A(z)A(z^{-1})$ and thus equation 6.3 follows.□

The discrete Fourier transform on a Golay sequence $A = (A_0, A_1, \ldots, A_{N-1})$ yields an infinite complex sequence, DFT_{A_ω}:

$$DFT_{A_\omega} = \sum_{j=0}^{N-1} A_j \omega^j = A(\omega) \tag{6.4}$$

where $|\omega| = 1$.

Corollary 2. *A pair* (A, B) *of length* N *is called a Golay binary pair if*

$$|DFT_{A_\omega}|^2 + |DFT_{B_\omega}|^2 = 2N, \quad \forall \omega, |\omega| = 1. \tag{6.5}$$

Proof:

We have that, $|\omega^k| = 1$ and therefore, from 6.3, $|DFT_{A_\omega}|^2 + |DFT_{B_\omega}|^2 = |A(\omega)|^2 + |B(\omega)|^2 = 2N$.□

Remark:

When $\omega = e^{k\frac{2\pi i}{N}}$, where $i = \sqrt{-1}$, we recover the kth point of the periodic discrete Fourier transform, but this is only a special case of the more general property where $|\omega| = 1$. Thus, 5.5 identifies that the pair (A, B) is complementary with respect to the continuous discrete Fourier transform.

6.1.2 Equivalence and Constructions of Golay binary sequences

Equivalence classes of Golay binary sequences

The following operations on a Golay pair (A, B) of length N preserve their length and complementarity [14] and form the equivalence class of (A, B). The output pair (A', B') of length N of each operation forms one of the elements in the equivalence class of (A, B),

1. Exchanging A and B in the pair.
2. Reversing the order of either A or B or both, *i.e.* $(A', B') = (\overleftarrow{A}, B)$, where $\overleftarrow{A} = (A_{N-1}, \ldots, A_1, A_0)$ if $A = (A_0, A_1, \ldots, A_{N-1})$.
3. Multiplying either A, B or both of them by -1.
4. The linear offset transformation, $(A', B') = ((-1)^i A_i, (-1)^i B_i)$.

Constructions of Golay binary sequences

By simple algebraic operations on equation 6.3, many recursive constructions for Golay sequences can be obtained. The following are the well-known recursive constructions:

1. The Golay-Shapiro-Rudin [31, 14] recursion generates a length $2N$ Golay sequence from a length N Golay sequence. It can be stated as follows. Let (A, B) be a Golay pair of length N, then simple algebraic manipulation shows that $A(z) + z^N B(z)$ and $A(z) - z^N B(z)$ satisfy equation 6.3. This could also be written in terms of A and B as:

$$(A, B) \rightarrow (A|B, A|-B) \text{ where '|' means concatenation.}$$

2. Turyn's construction [41] can be stated as follows. Let (C, D) and (A, B) be Golay pairs of lengths M and N respectively. Then

$$C(z^N)(A(z) + B(z))/2 + z^{N(M-1)} D(z^{-N})(A(z) - B(z))/2,$$
$$D(z^N)(A(z) + B(z))/2 - z^{N(M-1)} C(z^{-N})(A(z) - B(z))/2$$

is a Golay pair of length MN.

3. Golay's concatenation and interleaving constructions [14] are stated as follows. Let $(C, \overleftarrow{D})$ and (A, B) be Golay pairs of lengths M and N respectively, where $\overleftarrow{D}$ means reversal of D. Then Golay's concatenation construction is stated as follows

$$C(z^N)A(z) + D(z^N)B(z)z^{MN} \text{ and } \overleftarrow{D}(z^N)A(z) - \overleftarrow{D}(z^N)B(z)z^{MN}$$

and Golay's interleaving construction is stated as follows

$$C(z^{2N})A(z^2) + D(z^{2N})B(z^2)z \text{ and } \overleftarrow{D}(z^{2N})A(z^2) - \overleftarrow{D}(z^{2N})B(z^2)z$$

Given a fixed alphabet for the elements of our pair, we refer to a Golay pair as *primitive* if it cannot be constructed from shorter pairs over the same alphabet. Repeated application of Turyn's construction can be used to construct Golay pairs for all lengths $N = 2^a 10^b 26^c$ for all $a, b, c \geq 0$. For instance, we can generate an infinite number of pairs by starting with the following primitive pairs of lengths 2, 10 and 26:

(+ +, + -)

(- + + - + - + + + -, - + + + + + + - - +)

(+ - + - + + + + - -, + + + + - + + - - +)

(+ + + + - + + - - + - + - + - - + - + + + - - + + +, + + + + - + + - - + - + + + + + + - + - - - + + - - -)

where '+' and '−' mean 1 and −1 respectively.

In 1961, Golay [14] gave a direct construction for Golay pairs of length 2^m. In 1999, Davis and Jedwab gave an elegant description of this construction by using algebraic normal forms. Let

$$\begin{aligned} a(x_0, x_1, \ldots, x_{m-1}) &= \sum_{i=0}^{m-2} x_{\pi(i)} x_{\pi(i+1)} + \sum_{i=0}^{m-1} c_i x_i + c, \\ b(x_0, x_1, \ldots, x_{m-1}) &= \sum_{i=0}^{m-2} x_{\pi(i)} x_{\pi(i+1)} + \sum_{i=0}^{m-1} c_i x_i + x_{\pi(0)} + c' \end{aligned} \tag{6.6}$$

where π is a permutation of $\{0, 1, \ldots, m-1\}$ and c_i, c and $c' \in F_2$. Let $A_i = (-1)^{a(i_0,i_1,\ldots,i_{m-1})}$ and $B_i = (-1)^{b(i_0,i_1,\ldots,i_{m-1})}$, for $0 \leq i \leq 2^m - 1$ where $(i_0, i_1, \ldots, i_{m-1})$ is the binary representation of integer i. Then (A, B) is a Golay pair of length 2^m. Golay indicated [26] that this construction implies the existence of at least $m!2^m$ distinct Golay binary sequences of length 2^m.

6.2 Golay complementary array pairs

Let $A = (A[i_1, \ldots, i_r])$ be an r-dimensional array of size $s_1 \times \ldots \times s_r$ whose elements are defined over $S \subseteq \mathbb{C}$. The *aperiodic autocorrelation function* of an $s_1 \times \ldots \times s_r$ array $A = (A[i_1, \ldots, i_r])$ is given by [16]

$$C_A(k_1, \ldots, k_r) = \sum_{i_1=0}^{s_1-k_1-1} \ldots \sum_{i_r=0}^{s_r-k_r-1} A[i_1, \ldots, i_r]\overline{A[i_1 + k_1, \ldots, i_r + k_r]} \tag{6.7}$$

where $0 \leq k_q \leq s_q - 1$, $q \in \{1, \ldots, r\}$ and the bar represents complex conjugation. An $s_1 \times \ldots \times s_r$ *Golay complementary array pair* is a pair of $s_1 \times \ldots \times s_r$ arrays A and B over $S \subseteq \mathbb{C}$, for which

$$C_A(k_1, \ldots, k_r) + C_B(k_1, \ldots, k_r) = 0 \text{ for all } (k_1, \ldots, k_r) \neq 0.$$

We call an array A a *Golay array* if it forms a Golay array pair with some array B.

The following theorems summarize the known results about the existence of Golay arrays.

Theorem 23. *[19] If there exist binary Golay sequence pairs of length $s_1, \ldots, s_r$ then there exists an $s_1 \times \ldots \times s_r$ binary Golay array pair.*

Corollary 3. *[16] There exists an $s_1 \times \ldots \times s_r$ binary Golay array pair, where each s_k takes the form $2^{a_k}10^{b_k}26^{c_k}$ for integer a_k, b_k, c_k.*

In 1992, Dymond [10] proved an important generalization of Theorem 19 to the construction of multidimensional arrays.

Theorem 24. *[10] If there exist Golay pairs of size $s_1 \times \ldots \times s_r$ and $t_1 \times \ldots \times t_r$ then there exists a Golay pair of size $s_1 t_1 \times \ldots \times s_r t_r$.*

Jedwab and Parker indicated [16] that Theorem 23 can be recovered by repeated application of Dymond's multidimensional construction (note that we have not stated the multidimensional construction in Theorem 24, for further information about the construction refer to [10]).

Reference [16] proved that there is a relation between the aperiodic autocorrelations of a $(d+1)$-dimensional array and a d-dimensional array. The following lemma illustrates this relation.

Lemma 1. *[16] For integer $r \geq 0$, let $A = (A[i, j, i_1, \ldots, i_r])$ be an $s \times t \times s_1 \times \cdots \times s_r$ array. Define the $st \times s_1 \times \cdots s_r$ array $\psi(A) = (B[m, i_1, \ldots, i_r])$ by $B[ti + j, i_1, \ldots, i_r] := A[i, j, i_1, \ldots, i_r]$ for $0 \leq i < s, 0 \leq j < t, 0 \leq i_k < s_k (k = 1, \ldots, r)$. Then, for all integer $u, v, u_1, \ldots, u_r$, where $0 \leq v < t$,*

$$C_{\psi(A)}(tu + v, u_1, \ldots, u_r) = C_A(u, v, u_1, \ldots, u_r) + C_A(u + 1, v - t, u_1, \ldots, u_r).$$

One can use Lemma 1 to show that the existence of a $(d+1)$-dimensional array pair implies the existence of a d-dimensional array pair.

Theorem 25. *[16] For integer $r \geq 0$, suppose that A and B form an $s \times t \times s_1 \times \cdots \times s_r$ Golay array pair over an alphabet S. Then $\psi(A)$ and $\psi(B)$, as defined in Lemma 1, form an $st \times s_1 \times \cdots \times s_r$ Golay array pair over S.*

Proof:[16]

Fix integers $u, v, u_1, \ldots, u_r$ where $(u, v, u_1, \ldots, u_r) \neq 0$. By Lemma 1,

$$C_{\psi(A)}(tu + v, u_1, \ldots, u_r) + C_{\psi(B)}(tu + v, u_1, \ldots, u_r) = C_A(u, v, u_1, \ldots, u_r) + C_A(u + 1, v - t, u_1, \ldots, u_r) + C_B(u, v, u_1, \ldots, u_r) + C_B(u + 1, v - t, u_1, \ldots, u_r) = 0,$$

since A and B form a Golay pair. Therefore $\psi(A)$ and $\psi(B)$ form a Golay array pair over S.□

Repeated application of Theorem 26 gives us the following Corollary.

Corollary 4. *[16] If there exists an $s_1 \times \ldots \times s_r$ Golay array pair then there exists a Golay sequence pair of length $\prod_{k=1}^{r} s_k$ over the same alphabet.*

6.2.1 Spectral property of Golay array pairs

A Golay array $A = A[i_1, \ldots, i_r]$ of size $N_1 \times \ldots \times N_r$ can be associated with a polynomial

$$A(z_1, \ldots, z_r) = \sum_{i_1=0}^{N_1-1} \ldots \sum_{i_r=0}^{N_r-1} A[i_1, \ldots, i_r] z_1^{i_1} \cdots z_r^{i_r}$$

in indeterminates $(z_1, \ldots, z_r) \neq 0$ with coefficients in $\mathbb{C}$. Saying that the associated polynomials of $(A(z_1, \ldots, z_r), B(z_1, \ldots, z_r))$ are complementary shall mean that the pair (A, B) is complementary, where A and B are sequences whose elements are the coefficients of $A(z_1, \ldots, z_r)$ and $B(z_1, \ldots, z_r)$ respectively. The following theorem generalizes Theorem 22.

Theorem 26. *[12] An array pair (A, B) of size $N_1 \times \ldots \times N_r$ is called a Golay pair if the associated polynomials $(A(z_1, \ldots, z_r), B(z_1, \ldots, z_r))$ satisfy*

$$A(z_1, \ldots, z_r)\overline{A(z_1^{-1}, \ldots, z_r^{-1})} + B(z_1, \ldots, z_r)\overline{B(z_1^{-1}, \ldots, z_r^{-1})} = \epsilon(A) + \epsilon(B)$$

Moreover, if $z_1, \ldots, z_r$ are restricted to lie on the unit circle in the complex plane. Then

$$|A(z_1, \ldots, z_r)|^2 + |B(z_1, \ldots, z_r)|^2 = \epsilon(A) + \epsilon(B)$$

where $\epsilon(A) = \sum_{i_1=1}^{N_1-1} \ldots \sum_{i_r=0}^{N_r-1} |A[i_1, \ldots, i_r]|^2$ ($\epsilon(B)$ is defined similarly).
Proof:[12]

Straightforward manipulation shows that

$$A(z_1, \ldots, z_r)\overline{A(z_1^{-1}, \ldots, z_r^{-1})} = \sum_{k_1} \ldots \sum_{k_r} C_A(k_1, \ldots, k_r)((z_1^{k_1} \cdots z_r^{k_r}) + (z_1^{-k_1} \cdots z_r^{-k_r}))$$

Since $C_A(k_1, \ldots, k_r) + C_B(k_1, \ldots, k_r) = 0$ for all $(k_1, \ldots, k_r) \neq 0$, then $A(z_1, \ldots, z_r)\overline{A(z_1^{-1}, \ldots, z_r^{-1})} + B(z_1, \ldots, z_r)\overline{B(z_1^{-1}, \ldots, z_r^{-1})} = C_A(0, \ldots, 0) + C_B(0, \ldots, 0) = \epsilon(A) + \epsilon(B)$. If $|z_i| = 1$ for $1 \leq i \leq r$, then $|A(z_1, \ldots, z_r)|^2 = A(z_1, \ldots, z_r)\overline{A(z_1^{-1}, \ldots, z_r^{-1})}$ and therefore the second equation follows. $\square$

The spectral property of a Golay array can be interpreted by the multidimensional discrete Fourier transform. The multidimensional discrete Fourier transform of an $N_1 \times \ldots \times N_r$ Golay array $A = (A[i_1, \ldots, i_r])$ yields the infinite set of complex values $DFT_{A_{\omega_1,\ldots,\omega_r}}$:

$$DFT_{A_{\omega_1,\ldots,\omega_r}} = \sum_{i_1=0}^{N_1-1} \ldots \sum_{i_r=0}^{N_r-1} \omega_1^{i_1} \cdots \omega_r^{i_r} A[i_1, \ldots, i_r] = A(\omega_1, \ldots, \omega_r) \tag{6.8}$$

where $|\omega_j| = 1$, for $1 \leq j \leq r$.

The following Corollary generalizes Corollary 2,

Corollary 5. *A pair (A, B) of size $N_1 \times \ldots \times N_r$ is called an array pair if*

$$|DFT_{A_{\omega_1,\ldots,\omega_r}}|^2 + |DFT_{B_{\omega_1,\ldots,\omega_r}}|^2 = \epsilon(A) + \epsilon(B)$$

where $\omega_j = 1$, for $1 \leq j \leq r$ and $\epsilon(A)$ is defined as mentioned previously.

Proof:

The proof is similarly to the proof of Corollary 2. □

Remark:

When $\omega_j = e^{k_j \frac{2\pi i}{N_j}}$, $0 \leq k_j \leq N_j - 1$, for $1 \leq j \leq r$, we recover a point of the multidimensional periodic Fourier transform but this is only a special case of the more general property where $|\omega_j| = 1$. Thus Corollary 5 identifies that the array pair (A, B) is complementary with respect to the continuous multidimensional discrete Fourier transform.

6.2.2 Constructions of Golay arrays

Borwein and Ferguson [3] generalized Turyn's construction as follows.

Theorem 27. *[3] Suppose each of the pairs $(C(r), D(r))$ and $(A(s), B(s))$ is complementary. Then the pair*

$$F(r, s, t, u) = C(r)A(s) + \overline{D(r^{-1})}B(s)t,$$
$$G(r, s, t, u) = (D(r)A(s) + \overline{C(r^{-1})}B(s)t)u$$

is complementary.

Proof:[3]

According to Theorem 26, the pair $(F(r,s,t,u), G(r,s,t,u))$ is complementary if

$$F(r,s,t,u)\overline{F(r^{-1},s^{-1},t^{-1},u^{-1})} + G(r,s,t,u)\overline{G(r^{-1},s^{-1},t^{-1},u^{-1})}$$

is constant. Substituting,

$$F(r,s,t,u)\overline{F(r^{-1},s^{-1},t^{-1},u^{-1})} + G(r,s,t,u)\overline{G(r^{-1},s^{-1},t^{-1},u^{-1})} =$$
$$(C(r)\overline{C(r^{-1})} + D(r)\overline{D(r^{-1})})(A(s)\overline{A(s^{-1})} + B(s)\overline{B(s^{-1})})$$

which is constant since A and B is a complementary pair. □

By making appropriate substitutions in the above construction, the following constructions can be derived:

1. If (A, B) is complementary and $C(r) = D(r) = t = u = 1$, then $(F, G) = (A+B, A-B)$.

2. Turyn's construction mentioned in section 6.1.2 can be derived by starting with the complementary pairs $(C(r), D(r))$ and $((A(s)+B(s))/2, (A(s)-B(s))/2)$ and setting $r = z^{N_s}, s = z, t = z^{N_s(N_r-1)}$ and $u = 1$.

The construction in Theorem 27 can be generalized to the following multidimensional version of Turyn's construction.

Theorem 28. *Abbreviate $(s_0, \ldots, s_{n_s-1})$ and $(r_0, \ldots, r_{n_r-1})$ to s and r respectively. Let r, s and t be indeterminates. Let $(A(s), B(s))$ be the associated polynomials of a complementary array pair (A, B) of size $i_0 \times \ldots \times i_{n_s-1}$, (A, B). Let $(C(r), D(r))$ be the associated polynomials of a complementary array pair (C, D) of size $j_0 \times \ldots \times j_{n_r-1}$ then the array pair of size $i_0 \times \ldots \times i_{n_s-1} \times j_0 \times \ldots \times j_{n_r-1} \times 2$ having the following associated polynomials*

$$F(r,s,t) = C(r)A(s) + D^*(r)B(s)t,$$
$$G(r,s,t) = D(r)A(s) - C^*(r)B(s)t$$

is complementary, where $D^*(r) = (r_0^{j_0-1} \cdots r_{n_r-1}^{j_{n_r-1}-1})\overline{D(r_0^{-1}, \ldots, r_{n_r-1}^{-1})}$.

Proof:

According to Theorem 26, $(F(r,s,t), G(r,s,t))$ is complementary if

$$F(r,s,t)\overline{F(r^{-1},s^{-1},t^{-1})} + G(r,s,t)\overline{G(r^{-1},s^{-1},t^{-1})}$$

is constant. Substituting,

$$F(r,s,t)\overline{F(r^{-1},s^{-1},t^{-1})} + G(r,s,t)\overline{G(r^{-1},s^{-1},t^{-1})} =$$
$$(C(r)\overline{C(r^{-1})} + D(r)\overline{D(r^{-1})}(A(s)\overline{A(s^{-1})} + B(s)\overline{B(s^{-1})}) = (\epsilon(C) + \epsilon(D))(\epsilon(A) + \epsilon(B)),$$

which is constant. $\square$

Chapter 7

Type-I/II/III Pairs

7.1 Introduction

In this chapter we rename the Golay array pair construction, where each dimension is of size 2 as, the Type-I construction. We then introduce two new variants of the Golay array construction, naming them Type-II and Type-III complementary constructions.

Throughout this chapter, the term "an array of size 2^n" refers to an n-dimensional array of size $2 \times \ldots \times 2$. Let $V = (V_{00\ldots0}, V_{00\ldots01}, \ldots, V_{11\ldots1})$ be an array of size 2^n and U be a matrix of size $2^n \times 2^n$ which is a tensor product of 2×2 unitary matrices. Then UV is an array of size 2^n. $|UV|$ refers to the array obtained by taking the absolute value of each entry of UV.

The spectral property of an n-dimensional Golay array of size 2^n can be re-expressed using a $2^n \times 2^n$ unitary matrix. The following theorem illustrates this re-expression,

Theorem 29. *Let F and G be a Golay array pair of size 2^n such that $\epsilon(F) = \epsilon(G) = 2^n$. Then the spectral property of F and G is expressed as follows*

$$|F(z_0, z_1, \ldots, z_{n-1})|^2 + |G(z_0, z_1, \ldots, z_{n-1})|^2 = 2 \times 2^n, \tag{7.1}$$

where $|z_j| = 1$ for $0 \leq j \leq n-1$ and $\epsilon(F)$ is as defined previously in Theorem 26.
Let $U_I = U_0 \otimes U_1 \otimes \cdots \otimes U_{n-1}$, where

$$U_j = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \alpha_j \\ 1 & -\alpha_j \end{pmatrix} \tag{7.2}$$

for some arbitrary $\alpha_j \in \mathbb{C}$, where $|\alpha_j| = 1$ for all $0 \leq j \leq n-1$. Let $\hat{F} = U_I F$ and

$\hat{G} = U_I G$. *Then the above spectral property 7.1 can be re-expressed as follows:*

$$|\hat{F}_{k_0,\ldots,k_{n-1}}|^2 + |\hat{G}_{k_0,\ldots,k_{n-1}}|^2 = 2 \tag{7.3}$$

where $\hat{F}_{k_0,\ldots,k_{n-1}}$ *is the entry at index* $(k_0,\ldots,k_{n-1}) \in F_2^n$ *in* $\hat{F} = U_I F$.
Proof:

The proof follows immediately from Corollary 5. □

Let us call a matrix U_I formed from a tensor product of 2×2 matrices of the type specified in 7.2 a *Type-I* unitary matrix (Each U_j is unitary and therefore U_I is also unitary). Thus an array pair (F, G) is *Type-I complementary* if it satisfies 7.3. Theorem 29 says that any Golay array pair of size 2^n is a Type-I complementary pair and the converse is also true. Thus a "Type-I complementary pair" is just a new name for a Golay pair of size 2^n. The complementarity obtained by U_I raises the question, does the complementarity property hold for other types of unitary matrices ? Generally, a 2×2 unitary matrix can be written as,

$$u = D \begin{pmatrix} \cos(\theta) & \sin(\theta)e^{i\phi} \\ \sin(\theta) & -\cos(\theta)e^{i\phi} \end{pmatrix} \tag{7.4}$$

where θ and ϕ are arbitrary angles, and D is a diagonal or anti-diagonal 2×2 unitary matrix. θ and ϕ can be viewed as defining points on the surface of a sphere which is sometimes called the "Bloch sphere". Type-I unitaries, U_I, are constructed from 2×2 unitaries where $\theta = \frac{\pi}{4}$. Two other types of unitary matrices, named *Type-II* and *Type-III*, can be identified by assigning $\phi = 0$ and $\phi = \frac{\pi}{2}$ in 7.4 respectively. Now, we have the following three types of unitary matrix representing the 3 circumferences of the sphere that meet at right-angles:

Let $N = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}$, where $i = \sqrt{-1}$. Then the following relations hold:

$$u_I = D u_{III} N, \quad u_{II} = D' u_I N \quad \textit{and} \quad u_{III} = D'' u_{II} N \tag{7.5}$$

$$
\begin{aligned}
\text{Type-I:} \quad u_I &= \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & e^{i\phi} \\ 1 & -e^{i\phi} \end{pmatrix}, \\
\text{Type-II:} \quad u_{II} &= \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{pmatrix}, \\
\text{Type-III:} \quad u_{III} &= \begin{pmatrix} \cos(\theta) & \sin(\theta)i \\ \sin(\theta) & -\cos(\theta)i \end{pmatrix}
\end{aligned}
$$

where D, D' and D'' are arbitrary 2×2 diagonal or anti-diagonal unitary matrices. The complementarity in Type-II and Type-III is defined in the same way as the complementarity in Type-I. Let U_I, U_{II} and U_{III} be the unitary matrices formed from the tensor products of Type-I, Type-II and Type-III unitaries respectively. An array pair (F, G) of size 2^n is Type-I, Type-II or Type-III complementary if

$$
|\hat{F}_{k_0,\ldots,k_{n-1}}|^2 + |\hat{G}_{k_0,\ldots,k_{n-1}}|^2 = c \tag{7.6}
$$

where $\hat{F}_{k_0,\ldots,k_{n-1}}$ is the entry at index $(k_0, \ldots, k_{n-1}) \in F_2^n$ in $\hat{F} = UF$, c is a constant and $U = U_I, U = U_{II}$ or $U = U_{III}$ respectively. If $\epsilon(F) = \epsilon(G) = 2^n$, then c equals 2. Let $(F = (-1)^f, G = (-1)^g)$ be a complementary binary array pair of size 2^n, where f and g are n-variable Boolean functions, then we say that the Boolean function pair (f, g) is complementary.

Peak-to-average power ratio of arrays of size 2^n

The *peak-to-average power ratio* (PAPR) of an n-dimensional array of size $2 \times \cdots \times 2$, V, with respect to a set of unitary matrices, J, denoted by $P_J(V)$ is defined as

$$
P_J(V) = \max_{U \in J} \frac{2^n |UV|^2}{\sum_{y \in F_2^n} |V(y)|^2} \tag{7.7}
$$

Thus, $1 \leq P_J(V) \leq 2^n$. For a given set of unitary matrices, J, we are interested in constructing arrays, V, such that $P_J(V)$ is as small as possible. It is difficult to construct such arrays but as we will see in section 7.5, it is easier to construct pairs of arrays such

that their power sum, $P_J(V, V')$, is as small as possible, where

$$P_J(V, V') = \max_{U \in J} \frac{2^n(|UV|^2 + |UV'|^2)}{\sum_{y \in F_2^n} |V(y)|^2 + |V'(y)|^2} \tag{7.8}$$

For a pair of arrays (V, V'),

$$P_J(V) \leq \frac{\sum_{y \in F_2^n}(|V(y)|^2 + |V'(y)|^2)}{\sum_{y \in F_2^n} |V(y)|^2} P_J(V, V') \tag{7.9}$$

Thus, constructing pairs also provides a tight bound on each member of the pair. In this Chapter, we consider the case, $\epsilon(V) = \sum_{y \in F_2^n} |V(y)|^2 = \epsilon(V') = \sum_{y \in F_2^n} |V'(y)|^2$, which gives us the following upper bound,

$$P_J(V) \leq 2P_J(V, V'). \tag{7.10}$$

Corollary 6. *Let V be an array of size 2^n with $\epsilon(V) = 2^n$. Then the peak-to-average power ratio of V with respect to Type-I , Type-II or Type-III matrices is at most 2 if V is a Type-I, Type-II or Type-III member of an array pair respectively.*
Proof:

We prove the Type-I case. The other cases being similar. Let V be a member of a Type-I array pair, Let $P_I(V)$ denote the peak-to-average power ratio with respect to Type-I unitary matrices. Then from relations 7.6 and 7.7, $P_I(V) = \max |U_I V|^2 \leq 2$.□
If V is a binary array of size 2^n, then V can be written as $(-1)^v$ where v is an n-variable Boolean function. In this Chapter, the PAPR of a Boolean function v shall mean the PAPR of $(-1)^v$, *i.e.*, $P_J(v) \equiv P_J(V)$. The PAPR of two Boolean functions v and v' shall also mean the PAPR of $(-1)^v$ and $(-1)^{v'}$, *i.e.*, $P_J(v, v') \equiv P_J((-1)^v, (-1)^{v'})$.

Near-complementary pairs

It is of interest to construct pairs which are as near-complementary[1] as possible with respect to the action of a $2^n \times 2^n$ unitary, U, which is an n-fold tensor product of any 2×2. An

[1]Near-complementary pairs were constructed by Parker and Tellambura in [27] and also by Schmidt in [36].

array pair (F, G) of size 2^n is *near-complementary* with respect to a set of unitary matrices, J, if $P_J(F, G)$ is minimal or close to minimal for elements of F and G taken from a fixed alphabet. If J is the set of Type-I/II/III matrices, then (F, G) is a Type-I/II/III near complementary pair respectively. From equations 7.8 and 7.10, we see that $P_J(F, G)$ is an upper bound for the PAPR of either F or G. Thus constructing good near complementary pairs can be used as a construction for arrays with low PAPR.

Conversions between Type-I/II/III complementary array pairs

By using the relations in 7.5, Type-X can be converted to Type-Y, for X and Y taken from I, II, and III. The following theorem illustrates the conversions among the three types,

Theorem 30. *Let $(A_I, B_I), (A_{II}, B_{II})$ and (A_{III}, B_{III}) be Type-I, Type-II and Type-III complementary pairs respectively. Then the conversions of (A_I, B_I): $(N^{2^{\otimes n}} A_I, N^{2^{\otimes n}} B_I)$ and $(N^{\otimes n} A_I, N^{\otimes n} B_I)$, are Type-II and Type-III complementary pairs respectively, the conversions of (A_{II}, B_{II}): $(N^{\otimes n} A_{II}, N^{\otimes n} B_{II})$ and $(N^{2^{\otimes n}} A_{II}, N^{2^{\otimes n}} B_{II})$, are Type-I and Type-III complementary pairs respectively, and the conversions of (A_{III}, B_{III}): $(N^{2^{\otimes n}} A_{III}, N^{2^{\otimes n}} B_{III})$ and $(N^{\otimes n} A_{III}, N^{\otimes n} B_{III})$, are Type-I and Type-II complementary pairs respectively.*
Proof:

We prove the statement for the conversions of (A_I, B_I). The other proofs are similar. Let U_I, U_{II} and U_{III} be the unitary matrices formed from tensor product of Type-I, Type-II and Type-III unitaries respectively. We know that (A_I, B_I) is Type-I complementary if $|U_I A_I|^2 + |U_I B_I|^2 = 2$. From 7.5, we see that $U_I = u_I^{\otimes n} = (Du_{III})^{\otimes n} = D_1 U_{III} N^{\otimes n}$ and also see that $U_I = D_2 U_{II} N^{2^{\otimes n}}$ (since N^3 is a diagonal matrix). Thus,

$$|U_I A_I|^2 + |U_I B_I|^2 = |D_2 U_{II} N^{2^{\otimes n}} A_I|^2 + |D_2 U_{II} N^{2^{\otimes n}} B_I|^2 = |D_1 U_{III} N^{\otimes n} A_I|^2 +$$
$$|D_1 U_{III} N^{\otimes n} B_I|^2 = |U_{II} N^{2^{\otimes n}} A_I|^2 + |U_{II} N^{2^{\otimes n}} B_I|^2 = |U_{III} N^{\otimes n} A_I|^2 + |U_{III} N^{\otimes n} B_I|^2 = 2$$

and therefore the conversions of (A_I, B_I), $(N^{2^{\otimes n}} A_I, N^{2^{\otimes n}} B_I)$ and $(N^{\otimes n} A_I, N^{\otimes n} B_I)$ are Type-II and Type-III complementary pairs respectively. □

Remark:

In the above theorem, replacing N^2 by N^{-1} will also be a valid alternative since $N^2 = DN^{-1}$ where D is a 2×2 diagonal matrix. Observe that converting a Type-I binary pair to Type-II or Type-III will not preserve the binary alphabet of the pair in Type-II or Type-III (See Examples 1 and 2 on pages 76 and 77). In section 7.2, we show how to convert the Type-I construction itself in order to facilitate the construction of binary pairs in Type-II and Type-III.

Symmetry properties in Type-I/II/III

The following transformations on a Type-I/II/III complementary array pair (F, G) of size 2^n preserve the size and complementarity of (F, G):

1. $(F, G) \rightarrow (F' = F, G' = -G)$.
2. $(F = (-1)^f, G = (-1)^g) \rightarrow (F' = (-1)^{f+l}, G' = (-1)^{g+l})$ where f and g are n-variable Boolean functions and l is an n-variable affine function.
3. $(F, G) \rightarrow (\beta_1 F + \beta_2 G, \beta_3 F + \beta_4 G)$ where $\beta_1, \beta_2, \beta_3$ and β_4 form the 2×2 unitary matrix $U = \begin{pmatrix} \beta_1 & \beta_3 \\ \beta_2 & \beta_4 \end{pmatrix}$.

The proof of property 1 is trivial. Property 2 can be recovered be repeated application of property 1. Property 3 can be proved as follows: we prove the case for Type-I. The other cases follow similarly. Let $F(z_0, \ldots, z_{n-1})$ and $G(z_0, \ldots, z_{n-1})$ be the associated polynomials of F and G respectively. $(\beta_1 F + \beta_2 G, \beta_3 F + \beta_4 G)$ is Type-I complementary if

$$(\beta_1 F(z_0, \ldots, z_{n-1}) + \beta_2 G(z_0, \ldots, z_{n-1}))\overline{(\beta_1 F(z_0^{-1}, \ldots, z_{n-1}^{-1}) + \beta_2 G(z_0^{-1}, \ldots, z_{n-1}^{-1}))} +$$
$$(\beta_3 F(z_0, \ldots, z_{n-1}) + \beta_4 G(z_0, \ldots, z_{n-1}))\overline{(\beta_3 F(z_0^{-1}, \ldots, z_{n-1}^{-1}) + \beta_4 G(z_0^{-1}, \ldots, z_{n-1}^{-1}))}$$

is constant. Now, since U is a unitary matrix then $UU^* = I$, where U^* is the conjugate transpose of matrix U. Therefore, $\beta_1\overline{\beta_2} + \beta_3\overline{\beta_4} = \beta_2\overline{\beta_1} + \beta_4\overline{\beta_3} = 0$ and $\beta_1\overline{\beta_1} + \beta_3\overline{\beta_3} = \beta_2\overline{\beta_2} + \beta_4\overline{\beta_4} = 0$. Evaluating the above expression and using the above identities, we get

$$F(z_0,\ldots,z_{n-1})\overline{F(z_0^{-1},\ldots,z_{n-1}^{-1})}+G(z_0^{-1},\ldots,z_{n-1}^{-1})\overline{G(z_0^{-1},\ldots,z_{n-1}^{-1})}$$

which is constant by our hypothesis. Thus $(\beta_1 F+\beta_2 G, \beta_3 F+\beta_4 G)$ is Type-I complementary.□

Remark: Property 1 is a special case of property 3 as it can be obtained by setting $\beta_1 = 1$, $\beta_2 = \beta_3 = 0$ and $\beta_4 = -1$. One can also show that property 2 is a special case of the repeated application of property 3.

7.2 Type-I/II/III Constructions

We know that Type-I complementary pairs are Golay array pairs of size 2^n and that immediately tells us that they can be constructed by one of the standard Golay array constructions, in contrast to both Type-II and Type-III which are not Golay pairs. This raises the question, how can we construct Type-II and Type-III complementary pairs directly as opposed to via rotation of Type-I pairs, as discussed in the previous section ?

7.2.1 Type-I Construction

As we noted above, Type-I pairs are Golay array pairs of size 2^n and therefore any method which constructs Golay array pairs of size 2^n is a construction method for Type-I. The following proposition shows how to construct Type-I pairs and generalizes to the construction of Type-I near-complementary pairs. It is just a special case of Theorem 28 (but generalizes Theorem 28 to near-complementarity).

Proposition 1. *Abbreviate $(s_0,\ldots,s_{n_s-1})$ and $(r_0,\ldots,r_{n_r-1})$ to s and r respectively. Let r, s and t be indeterminates. Let $(A(s), B(s))$ be the associated polynomials of a Type-I (near-)complementary array pair (A, B) of size 2^{n_s}. Let $(C(r), D(r))$ be the associated polynomials of a Type-I (near-)complementary array pair (C, D) of size 2^{n_r}. Then the array pair (F, G) of size $2^{n_s+n_r+1}$ having the associated polynomials,*

$$F(r,s,t) = C(r)A(s) + D^*(r)B(s)t \tag{7.11}$$

$$G(r,s,t) = D(r)A(s) - C^*(r)B(s)t \tag{7.12}$$

is Type-I (near-)complementary, where $D^*(r) = r_0 r_1 \cdots r_{n_r-1}\overline{D(r^{-1})}$ *and* $r^{-1} = (r_0^{-1}, r_1^{-1}, \ldots, r_{n_r-1}^{-1})$. *More precisely, if* r, s *and* t *are restricted to lie on the unit circle in the complex plane, then*

$$P_I(F, G) = P_I(A, B)P_I(C, D) \tag{7.13}$$

where P_I *means PAPR with respect to Type-I unitary matrices.*

Proof:

As in Theorem 28, evaluating $F(r, s, t)\overline{F(r^{-1}, s^{-1}, t^{-1})} + G(r, s, t)\overline{G(r^{-1}, s^{-1}, t^{-1})}$ gives us the constant $(C(r)\overline{C(r^{-1})} + D(r)\overline{D(r^{-1})})(A(s)\overline{A(s^{-1})} + B(s)\overline{B(s^{-1})})$ which proves that (F, G) is a Type-I complementary pair. Since t and the elements of r and s are restricted to lie on the unit circle in the complex plane, then $|F(r, s, t)|^2 + |G(r, s, t)|^2 = (|C(r)|^2 + |D(r)|^2)(|A(s)|^2 + |B(s)|^2))$.

Let $n = n_s + n_r + 1$. From Theorem 29, we see that $\max(|F(r, s, t)|^2 + |G(r, s, t)|^2) = \max(2^n(|U_I F|^2 + |U_I G|^2)) = 2^{n+1}P_I(F, G)$, $\max(|C(r)|^2 + |D(r)|^2) = 2^{n_r}\max(|U_I C|^2 + |U_I D|^2) = 2^{n_r+1}P_I(C, D)$ and $\max(|A(s)|^2 + |B(s)|^2) = 2^{n_s}\max(|U_I A|^2 + |U_I B|^2) = 2^{n_s+1}P_I(A, B)$. Combining all these equations together, we find that $P_I(F, G) = P_I(A, B)P_I(C, D)$.□

Remark:

Proposition 1 can be generalized for arrays of any combination of dimensions and not just for arrays where each dimension is of size 2.

Construction of Type-I binary pairs

We consider the construction of Type-I (near-)complementary binary arrays. Thus we assume in Proposition 1 that A, B, C and D are all binary arrays. So $D^*(r) = \overleftarrow{D}(r)$, where $\overleftarrow{D}(r)$ is the associated polynomial of $\overleftarrow{D}$ which is the reversal of D ($\overleftarrow{D}[i_1, \ldots, i_{n_r}] = D[1 - i_1, \ldots, 1 - i_{n_r}]$, where $(i_1, \ldots, i_{n_r}) \in F_2^{n_r}$). Moreover we shall consider only the case where $t = 1$. Thus we obtain from 7.11 and 7.12,

$$F(r, s) = C(r)A(s) + \overleftarrow{D}(r)B(s) \tag{7.14}$$

$$G(r, s) = D(r)A(s) - \overleftarrow{C}(r)B(s) \tag{7.15}$$

From symmetry property 3 we know that if (A, B) is Type-I complementary, then so is $((A+B)/2, (A-B)/2)$. Therefore, by replacing $A(s)$ and $B(s)$ with $((A(s)+B(s))/2)$ and $((A(s)-B(s))/2)$ respectively in equations 7.14 and 7.15, we arrive at the following complementary pair

$$F(r,s) = C(r)((A(s)+B(s))/2) + \overleftarrow{D}(r)((A(s)-B(s))/2) \tag{7.16}$$

$$G(r,s) = D(r)((A(s)+B(s))/2) - \overleftarrow{C}(r)((A(s)+B(s))/2) \tag{7.17}$$

The following Corollary proves that (F, G) is a binary pair and gives a direct formula for constructing binary pairs.

Corollary 7. *Let $(A = (-1)^a, B = (-1)^b)$ be a Type-I complementary binary array pair of size 2^{n_s}, where a and b are the ANFs of n_s-variable Boolean functions. Let $(C = (-1)^c, D = (-1)^d)$ be also a Type-I complementary binary array pair of size 2^{n_r}, where c and d are the ANFs of n_r-variable Boolean functions. Then the pair (F, G) in 7.16 and 7.17 is a binary pair and can be written as $(F = (-1)^f, G = (-1)^g)$, where f and g are given by*

$$\begin{aligned} f &= (a+b)(c+\overleftarrow{d}) + a + \overleftarrow{d} \\ g &= (a+b)(\overleftarrow{c}+d) + b + \overleftarrow{c} \end{aligned} \tag{7.18}$$

and $\overleftarrow{d}(r_0, r_1, \ldots, r_{n_r-1}) = d(r_0+1, r_1+1, \ldots, r_{n_r-1}+1)$, is Type-I (near-)complementary.

Proof:

We prove the formula for f. The proof for g is similar. The coefficient of $s_0^{x_0} \cdots s_{n_s-1}^{x_{n_s-1}} r_0^{x_{n_s}} \cdots r_{n_r-1}^{x_{n_s+n_r-1}}$, where $(x_0, \ldots, x_{n_s+n_r}) \in F_2^{n_s+n_r-1}$ in polynomial $F(r,s)$ in 7.16 is,

$$\tfrac{1}{2}(-1)^c((-1)^a + (-1)^b) + \tfrac{1}{2}(-1)^{\overleftarrow{d}}((-1)^a - (-1)^b)$$

Consider the subcases where $(a,b) = (0,0), (0,1), (1,0)$ and $(1,1)$, respectively. Thus, when $a = b = 0$, we obtain, the expression $(a+1)(b+1)c$. Similarly, when $a = 0, b = 1$, we obtain, $(a+1)b\overleftarrow{d}$. When $a = 1, b = 0$, we obtain, $a(b+1)(\overleftarrow{d}+1)$, and, when $a = b = 1$, we obtain, $ab(\overleftarrow{c}+1)$. By adding the four subexpressions together we find that $f = ac + bc + b\overleftarrow{d} + a\overleftarrow{d} + a + c$. Now if (c,d) is a pair then so is $(\overleftarrow{d}, \overleftarrow{c})$. Thus, replacing c by $\overleftarrow{d}$ and d by $\overleftarrow{c}$, in f, yields $f = a\overleftarrow{d} + b\overleftarrow{d} + bc + ac + a + d = (a+b)(c+\overleftarrow{d}) + a + \overleftarrow{d}$. □

7.2.2 Type-II Construction

Theorem 30 tell us how to convert Type-I pairs to Type-II pairs. We can exploit this idea to convert the Type-I construction over n variables to a Type-II construction by using the same approach to that used in Theorem 29, which is to left-multiply the Type-I construction by $N^{2^{\otimes n}}$ or by $N^{-1^{\otimes n}}$ (since $N^2 = DN^{-1}$, where D is a 2×2 diagonal matrix). The following proposition shows how to construct Type-II pairs.

Proposition 2. *Abbreviate $(s_0, \ldots, s_{n_s-1})$ and $(r_0, \ldots, r_{n_r-1})$ to s and r respectively. Let r, s and t be indeterminates. Let $(A(s), B(s))$ be the associated polynomials of a Type-I (near-)complementary array pair (A, B) of size 2^{n_s}. Let $(C(r), D(r))$ be the associated polynomials of a Type-I (near-)complementary array pair (C, D) of size 2^{n_r}. Then the array pair (F', G') of size $2^{n_s+n_r+1}$ having the associated polynomials,*

$$F'(r, s, t) = C'(r)A'(s) + \overline{D'(r)}B'(s) + i(\overline{D'(r)}B'(s) - C'(r)A'(s))t$$
$$G'(r, s, t) = D'(r)A'(s) - \overline{C'(r)}B'(s) - i(D'(r)A'(s) + \overline{C'(r)}B'(s))t,$$

is Type-II (near-)complementary, where $A' = (N^{-1})^{\otimes n_s}A, B' = (N^{-1})^{\otimes n_s}B, C' = (N^{-1})^{\otimes n_r}C, D' = (N^{-1})^{\otimes n_r}D$ (and therefore $\overline{D'} = (N^{-1})^{\otimes n_r}D^*)$[2], and $A(s), B(s), C(r)$ and $D(r)$ are the associated polynomials of arrays A, B, C and D respectively. More precisely,

$$P_{II}(F, G) = P_{II}(A', B')P_{II}(C', D') = P_I(A, B)P_I(C, D)$$

where P_{II} means PAPR with respect to Type-II unitary matrices.

Proof:

[2]The identity $\overline{D'} = (N^{-1})^{\otimes n_r}D^*$ in Proposition 2 can be proved as follows: Let $v = \begin{pmatrix} k+li \\ p+qi \end{pmatrix}$, where $k, l, p, q \in \mathbb{R}$ and $i = \sqrt{-1}$. Then, by definition, $v^* = \begin{pmatrix} p-qi \\ k-li \end{pmatrix}$. Let $v' = N^{-1}v$. Now, we want to prove that $\overline{v'} = N^{-1}v^*$. The right hand side (RHS), $N^{-1}v^* = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -i & i \end{pmatrix}\begin{pmatrix} p-qi \\ k-li \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} (k+p)-(l+q)i \\ (l-q)-(p-k)i \end{pmatrix}$. The left hand side (LHS), $\overline{v'} = \overline{\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -i & i \end{pmatrix}\begin{pmatrix} k+li \\ p+qi \end{pmatrix}} = \frac{1}{\sqrt{2}}\overline{\begin{pmatrix} (k+p)+(l+q)i \\ (l-q)+(p-k)i \end{pmatrix}}$ = RHS. Now the identity $\overline{D'} = (N^{-1})^{\otimes n_r}D^*$ follows by tensor expansion of the identity $\overline{v'} = N^{-1}v^*$.

Let $n = n_s + n_r + 1$. Let (F, G) be the array pairs of size 2^n having the associated polynomials in 7.11 and 7.12. Since the Type-II construction $= (N^2)^{\otimes n} = (N^{-1})^{\otimes n}$Type-I construction, then multiplying F and G by $(N^{-1})^{\otimes n}$, we obtain F' and G' respectively. We prove the formula for F'. The proof for G' is similar. $F' = (N^{-1})^{\otimes n} F = (N^{-1})^{\otimes n_s + n_r + 1} F = (I^{\otimes n_r + n_s} \otimes N^{-1})((N^{-1})^{\otimes n_s + n_r} \otimes I)F = (I^{\otimes n_r + n_s} \otimes N^{-1})\hat{F}$, where $\hat{F}$ is the array pair having the associated polynomial $C'(r)A'(s) + \overline{D'(r)}B'(s)t$. $(I^{\otimes n_r + n_s} \otimes N^{-1})\hat{F}$ gives us the array pair having the associated polynomial $C'(r)A'(s) + \overline{D'(r)}B'(s) + i(\overline{D'(r)}B'(s) - C'(r)A'(s))t$. □

Construction of Type-II binary pairs

We consider the construction of Type-II (near-)complementary binary arrays. By assuming in Proposition 2 that A', B', C' and D' are all binary arrays and that $t = 0$, so $\overline{D'(r)} = D'(r)$, we obtain the following construction for Type-II (near-)complementary pairs

$$F'(r, s) = C'(r)A'(s) + D'(r)B'(s) \tag{7.19}$$

$$G'(r, s) = D'(r)'A(s) - C'(r)B'(s) \tag{7.20}$$

From symmetry property 3 we know that if (A', B') is Type-II complementary, then so is $((A + B)/2, (A - B)/2)$. Therefore, by replacing $A'(s)$ and $B'(s)$ with $((A'(s) + B'(s))/2)$ and $((A'(s) - B'(s))/2)$ respectively in equations 7.19 and 7.20, we arrive at the following complementary pair

$$F'(r, s) = C'(r)((A'(s) + B'(s))/2) + D'(r)((A'(s) - B'(s))/2) \tag{7.21}$$

$$G'(r, s) = D'(r)((A'(s) + B'(s))/2) - C'(r)((A'(s) + B'(s))/2) \tag{7.22}$$

The following Corollary proves that (F', G') is a binary pair and gives a direct formula for constructing binary pairs.

Corollary 8. *Let $(A' = (-1)^a, B' = (-1)^b)$ be a Type-II binary complementary array pair of size 2^{n_s}, where a and b are the ANFs of n_s-variable Boolean functions. Let $(C' =$*

$(-1)^c, D' = (-1)^d)$ *be also a Type-II binary complementary array pair of size* 2^{n_r}*, where* c *and* d *are the ANFs of* n_r*-variable Boolean functions. Then the pair* (F, G) *in 7.21 and 7.22 is a binary pair and can be written as* $(F' = (-1)^f, G' = (-1)^g)$*, where* f *and* g*, given by*

$$\begin{aligned} f &= (a+b)(c+d) + a + d \\ g &= (a+b)(c+d) + b + c \end{aligned} \tag{7.23}$$

is Type-II (near-)complementary.

Proof:

The proof is similar to that for Corollary 8.□

7.2.3 Type-III Construction

One can use again the conversion idea behind the Type-II construction to find a construction for Type-III. Thus we convert the Type-I construction over n variables to a Type-III construction by left multiplying the Type-I construction by $N^{\otimes n}$, *i.e.* Type-III construction = $N^{\otimes n}$Type-I construction. The following proposition shows how to construct Type-III pairs.

Proposition 3. *Let* $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$*. Abbreviate* $(s_0, \ldots, s_{n_s-1})$ *and* $(r_0, \ldots, r_{n_r-1})$ *to* s *and* r *respectively. Let* r, s *and* t *be indeterminates. Let* $(A(s), B(s))$ *be the associated polynomials of a Type-I (near-)complementary array pair* (A, B) *of size* 2^{n_s}*. Let* $(C(r), D(r))$ *be the associated polynomials of a Type-I (near-)complementary array pair* (C, D) *of size* 2^{n_r}*. Then the array pair* (F'', G'') *of size* $2^{n_s+n_r+1}$ *having the associated polynomials,*

$$\begin{aligned} F''(r,s,t) &= C''(r)A''(s) + i^{n_r+1}D''_z(r)B''(s) + (C''(r)A''(s) - i^{n_r+1}D''_z(r)B''(s))t, \\ G''(r,s,t) &= D''(r)A''(s) - i^{n_r+1}C''_z(r)B''(s) + (D''(r)A''(s) + i^{n_r+1}C''_z(r)B''(s))t \end{aligned}$$

is Type-III (near-)complementary, where $A'' = N^{\otimes n_s}A, B'' = N^{\otimes n_s}B, C'' = N^{\otimes n_r}C, D'' = N^{\otimes n_r}D$ *(and therefore* $\overline{D''} = (-i)^{n_r}Z^{\otimes n_r}N^{\otimes n_r}D^*$*)*[3]*,* $C''_z(r) = Z^{\otimes n_r}\overline{C''}$*,* $D''_z(r) = Z^{\otimes n_r}\overline{D''}$*, and* $A(s), B(s), C(r)$ *and* $D(r)$ *are the associated polynomials of arrays* A, B, C *and* D *respectively. More precisely,* $P_{III}(F'', G'') = P_{III}(A'', B'')P_{III}(C'', D'') = P_I(A, B)P_I(C, D)$*.*
Proof:

The proof is similar to that for Type-II.

Construction of Type-III binary pairs

We consider the construction of Type-III (near-)complementary binary arrays. If (A'', B'') is a Type-III pair, using symmetry property 3, so is $(A'', i^k B'')$ for some integer k. By re-assigning B'' as $i^k B''$ for k chosen appropriately and assuming in Proposition 2 that A'', B'', C'' and D'' are all binary arrays and that $t = 0$, so $\overline{D''(r)} = D''(r)$, we obtain the following construction

$$F''(r, s) = C''(r)A''(s) + D''_z(r)B''(s) \tag{7.24}$$

$$G''(r, s) = D''(r)''A(s) - C''_z(r)B''(s) \tag{7.25}$$

From symmetry property 3 we know that, if (A'', B'') is Type-III complementary, then so is $((A+B)/2, (A-B)/2)$. Therefore, by replacing $A''(s)$ and $B''(s)$ with $((A''(s)+B''(s))/2)$ and $((A''(s)-B''(s))/2)$ respectively in equations 7.24 and 7.25, we arrive at the following

[3]The identity $\overline{D''} = (-i)^{n_r}Z^{\otimes n_r}N^{\otimes n_r}D^*$ in Proposition 3 can be proved as follows: Let $v = \begin{pmatrix} k+li \\ p+qi \end{pmatrix}$, where $k, l, p, q \in \mathbb{R}$ and $i = \sqrt{-1}$. Then, by definition, $v^* = \begin{pmatrix} p-qi \\ k-li \end{pmatrix}$. Let $v'' = Nv$. Now, we want to prove that $\overline{v''} = -iZNv^*$. The right hand side (RHS), $-iZNv^* = \frac{-i}{\sqrt{2}}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}\begin{pmatrix} p-qi \\ k-li \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} (k-q)-(p+l)i \\ (k+q)+(p-l)i \end{pmatrix}$. The left hand side (LHS), $\overline{v''} = \frac{1}{\sqrt{2}}\overline{\begin{pmatrix} 1 & i \\ 1 & -i \end{pmatrix}\begin{pmatrix} k+li \\ p+qi \end{pmatrix}} = \frac{1}{\sqrt{2}}\overline{\begin{pmatrix} (k-q)+(p+l)i \\ (k+q)-(p-l)i \end{pmatrix}}$ = RHS. Now the identity $\overline{D''} = (-i)^{n_r}Z^{\otimes n_r}N^{\otimes n_r}D^*$ follows by tensor expansion of the identity $\overline{v''} = -iZNv^*$.

complementary pair

$$F''(r,s) = C''(r)((A''(s) + B''(s))/2) + D''_z(r)((A''(s) - B''(s))/2) \tag{7.26}$$

$$G''(r,s) = D''(r)((A''(s) + B''(s))/2) - C''_z(r)((A''(s) + B''(s))/2) \tag{7.27}$$

The following Corollary proves that (F'', G'') is a binary pair and gives a direct formula for constructing binary pairs.

Corollary 9. *Let $(A'' = (-1)^a, B'' = (-1)^b)$ be a binary array pair of size 2^{n_s}, where a and b are the ANFs of n_s-variable Boolean functions. Let $(C'' = (-1)^c, D'' = (-1)^d)$ be also a binary array pair of size 2^{n_r}, where c and d are the ANFs of n_r-variable Boolean functions. Then the pair (F, G) in 7.26 and 7.27 is a binary pair and can be written as $(F'' = (-1)^f, G'' = (-1)^g)$, where f and g are given by*

$$\begin{aligned} f &= (a + b + l_r)(c + d) + a + d \\ g &= (a + b + l_r)(c + d) + b + c + l_r \end{aligned} \tag{7.28}$$

and l_r is the linear Boolean function $\sum_{i=n_s}^{n_s+n_r-1} x_i$, where $(x_{n_s}, \ldots, x_{n_s+n_r-1})$ are the Boolean variables of c and d, is Type-III (near-)complementary.
Proof:

The proof is similar to that for Corollary 8 (note that $(-1)^{l_r(x_{n_s},\ldots,x_{n_s+n_r-1})}$ corresponds to the diagonal entries of $Z^{\otimes n_r}$). □

7.3 Type-I/II/III complementary binary pairs

In this section, we use the constructions in 7.18, 7.23 and 7.28 to generate the ANF of Type-I, Type-II and Type-III binary complementary pairs. But in order to generate complementary pairs by these recursive constructions, we need initial binary complementary pairs to start with for each construction, *i.e.* primitive complementary pairs for Type-I, Type-II and Type-III. It is already known that the array pair of size 2, $(F = (-1)^f, G = (-1)^g)$,

where $f = 0$ and $g = x_0$, are Boolean functions defined over 1 variable, is a Type-I primitive complementary pair. We further determine that this pair is also a Type-II primitive complementary pair. In contrast to Type-I and Type-II, in Type-III, we found that any array pair of size 2^n consisting of any combination of affine functions is a Type-III complementary pair. Some of these affine array pairs are primitive and some are not. The array pairs $(F = (-1)^f, G = (-1)^g)$, where $(f = 0, g = 0)$, $(f = 0, g = x_0)$ and $(f = x_0, g = x_0)$ and f and g are defined over 1 variable, form the primitive array pairs of size 2. The array pair $(F = (-1)^f, G = (-1)^g)$, where $(f = \sum_{i=0}^{i=n-1} x_i, g = \sum_{i=0}^{i=n-1} x_i)$ is a primitive array pair of size 2^n. We use these primitive pairs as initial pairs to the Type-I and Type-II and Type-III binary constructions to construct complementary array pairs of larger size. Before describing the output of each construction, we need to introduce the following symmetry property for binary array pairs in Type-I/II/III: *If* $(F = (-1)^{f(x_0,\ldots,x_{n-1})}, G = (-1)^{g(x_0,\ldots,x_{n-1})})$*, where* f *and* g *are* n*-variable Boolean functions, is a complementary binary pair in Type-I/II/III, then so is* $(F' = (-1)^{f(x_{\pi(0)},\ldots,x_{\pi(n-1)})}, G' = (-1)^{g(x_{\pi(0)},\ldots,x_{\pi(n-1)})})$*, where* π *is any permutation of* $\{0, 1, \ldots, n-1\}$. We call this symmetry property the re-labeling property.

Type-I complementary binary pairs (Golay array pairs of size 2^n)

The primitive binary array pair for Type-I is $(F = (-1)^0, G = (-1)^{x_0})$. So we set $(a = 0, b = x_0)$ and $(c = 0, d = x_1)$ as initial pairs for the construction in 7.18. This outputs $f = x_0x_1 + x_0 + x_1 + 1$ and $g = x_0x_1 + x_0$, and thus $((-1)^f, (-1)^g)$ is an array pair of size 2^2. We then use $a = f$ and $b = g$, and $c = 0$ and $d = x_2$. This outputs $f = x_0x_1 + x_1x_2 + x_0 + 1$ and $g = x_0x_1 + x_1x_2 + x_0 + x_2$, and thus $((-1)^f, (-1)^g)$ is an array pair of size 2^3. Proceeding in this way, we notice that the construction in 7.18 generates only the array pair, $((-1)^f, (-1)^g)$, where $f = \sum_{i=0}^{n-2} x_ix_{i+1} + \sum_{i=2}^{n-2} x_i + x_0 + 1$ and $g = \sum_{i=0}^{n-2} x_ix_{i+1} + \sum_{i=2}^{n-1} x_i + x_0$. By using the symmetry properties 1 and 2, we find that $((-1)^f, (-1)^g)$, where $f = \sum_{i=0}^{n-2} x_ix_{i+1}$ and $g = f + x_{n-1}$ is a Type-I array pair (note that this is the same Golay sequence pair obtained in 6.6 by [8] and it was also obtained as an array pair in [12]). We use this pair as a canonical representation for all known Type-I array

pairs of size 2^n. Since f and g are quadratic functions, then Type-I complementary pairs can best be described by undirected graphs[4] as they simply form a path graph[5] (Type-I binary array pairs of size 2^n are Golay array pairs of size 2^n).

Type-II complementary binary pairs

The primitive binary array pair for Type-II is $(F = (-1)^0, G = (-1)^{x_0})$. So we set $(a = 0, b = x_0)$ and $(c = 0, d = x_1)$ as initial pairs for the construction in 7.23, this outputs $f = x_0x_1 + x_1$ and $g = x_0x_1 + x_0$, and thus $((-1)^f, (-1)^g)$ is an array pair of size 2^2. We then use $a = f$ and $b = g$, and $c = 0$ and $d = x_2$, this outputs $f = x_0x_1 + x_0x_2 + x_1x_2 + x_1 + x_2$ and $g = x_0x_1 + x_0x_2 + x_1x_2 + x_0$, and thus $((-1)^f, (-1)^g)$ is an array pair of size 2^3. Proceeding in this way, we notice that the construction in 7.23 generates only the array pair, $((-1)^f, (-1)^g)$, where $f = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} x_ix_j + \sum_{i=1}^{n-1} x_i$ and $g = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} x_ix_j + x_0$. By using the symmetry property 2, we find that $((-1)^f, (-1)^g)$, where $f = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} x_ix_j$ and $g = f + \sum_{i=0}^{n-1} x_i$ is an array pair. We use this pair as a canonical representation for all Type-II array pairs. Since f and g are quadratic functions, then Type-II complementary pairs can best be described by undirected graphs as they simply form a complete graph[6].

Type-III complementary binary pairs

Any array pair of size 2^n consisting of any combination of affine functions is a Type-III complementary pair. The following example shows that there are array pairs consisting of a combination of affine functions that are not primitive: Setting $(a = 0, b = x_0)$ and $(c = 0, d = 0)$ as initial pairs for the construction in 7.28, where (a, b) and (c, d) are defined over 1 variable, and thus $l_r = x_1$, outputs $f = 0$ and $g = x_0 + x_1$, and thus

[4] A graph is pair $G = (V, E)$, where $V = (v_0, v_1, \ldots, v_{n-1})$ is a set of n vertices (or nodes) and $E \subseteq V \times V$. A pair $v_i, v_j \in E$ is called an edge.

[5] A path graph is a graph with two nodes of vertex degree (the number of neighbors the node has) one and the other nodes of vertex degree 2.

[6] A complete graph is a graph where all pairs of vertices are connected by an edge.

$((-1)^f, (-1)^g)$ is an array pair of size 2^2. Using some combinations of these affine array pairs as (a, b) and (c, d) in 7.28, one can output many quadratic Boolean function pairs (f, g), which can be used again as either (a, b) or (c, d). This will generate an infinite number of quadratic Boolean function pairs (f, g) which give us an infinite number of array pairs $(F = (-1)^f, G = (-1)^g)$. For instance, setting $(a = 0,\ b = x_0)$, $(c = 0, d = x_3)$, where (a, b) are Boolean functions defined on 1 variable and (c, d) are Boolean functions defined over 3 variables and so $l_r = x_1 + x_2 + x_3$, outputs $f = x_0x_3 + x_1x_3 + x_2x_3$ and $g = f + x_0 + x_1 + x_2$, and thus $((-1)^f, (-1)^g)$ is an array pair of size 2^4. More generally if we set $(a = 0,\ b = x_0)$, $(c = 0, d = x_{n-1})$, where (a, b) are Boolean functions defined over 1 variable and (c, d) are Boolean functions defined over $n - 1$ variables and so $l_r = x_1 + x_2 + \ldots + x_{n-1}$, we get $f = x_0x_{n-1} + x_1x_{n-1} + \ldots + x_{n-2}x_{n-1}$ and $g = f + x_0 + x_1 + \ldots + x_{n-2}$, and thus $((-1)^f, (-1)^g)$ is an array pair of size 2^n. This pair is best described by a star graph[7].

Another star graph pair can be obtained as follows: setting $(a = 0, b = x_0)$ and $(c = 0, d = x_1)$ where (a, b) and (c, d) are both defined over 1 variable, outputs $f = x_0x_1$ and $g = x_0x_1 + x_0$. Then setting this (f, g) as (a, b) and $(c = 0, d = x_2)$ where (a, b) are defined over 2 variables and (c, d) are defined over 1 variable, outputs $f = x_0x_1 + x_0x_2$ and $g = f + x_0$. Then setting this (f, g) as (a, b) and $(c = 0, d = x_3)$ where (a, b) are defined over 3 variables and (c, d) are defined over 1 variable, outputs $f = x_0x_1 + x_0x_2 + x_0x_3$ and $g = f + x_0$. Proceeding in this way, after $n - 1$ steps, we arrive at $f = x_0x_1 + x_0x_2 + \ldots + x_0x_{n-1}$ and $g = f + x_0$. By re-labeling f and g (exchanging x_0 and x_{n-1}), we get the pair $f = x_0x_{n-1} + x_1x_{n-1} + \ldots + x_{n-2}x_{n-1}$ and $g = f + x_{n-1}$, which is a star graph similar to the star graph mentioned above but they only differ in the linear terms of g.

Now using the star graph $(f = x_0x_{n-2} + x_1x_{n-2} + \ldots + x_{n-3}x_{n-2}, g = f + x_0 + x_1 + \ldots + x_{n-3})$ as (a, b) where (a, b) are Boolean functions defined over $n - 1$ variables and setting $(c = 0, d = x_{n-1})$ where (c, d) are Boolean functions defined over 1 variable and so $l_r = x_{n-1}$,

[7] A star graph is a tree (a connected graph which does not contain a cycle) on n nodes with one node having vertex degree $n - 1$ and the other $n - 1$ having vertex degree 1.

outputs $f = x_0x_{n-2} + x_1x_{n-2} + \ldots + x_{n-3}x_{n-2} + x_0x_{n-1} + x_1x_{n-1} + \ldots + x_{n-3}x_{n-1}$ and $g = f + x_0 + \ldots + x_{n-3}$, and thus $((-1)^f, (-1)^g)$ is an array pair of size 2^n. So we see that the number of Type-III complementary array pairs of size 2^n grow as n grows, which is totally different from Type-I and Type-II as they have only one class of array pairs (the path graph and the complete graph respectively).

To characterize Type-III pairs generated by 7.28, we performed a search on connected quadratic Boolean functions[8] with number of variables between 2 and 7 to find Boolean functions with PAPR ≤ 2.0 with respect to Type-III matrices. We found many connected quadratic Boolean functions with PAPR ≤ 2.0 with respect to Type-III matrices. By adding a linear term for each function found, we found the other function that forms a pair with it. *i.e.* If f is a Boolean function with PAPR ≤ 2.0 with respect to Type-III matrices, then $(f, f + l)$, for some linear function l, is a Type-III complementary pair. We actually found two different pairs $(f, f + l_1)$ and $(f, f + l_2)$, where l_1 and l_2 are linear functions, for each function found f. The following proposition characterizes some of the connected complementary pairs in Type-III.

Proposition 4. *If* $f_1 = \sum_{i=0}^{j_0 \leq k-1} x_i x_k + \sum_{i=0}^{j_1 \leq k-1} x_i x_{k+1} + \cdots + \sum_{i=0}^{j_{n-k-1} \leq k-1} x_i x_{n-1}$ *(where* $k \leq n-1$ *and there is at least one* $j_s = k-1$ *where* $s \in \{0, 1, \ldots, n-k-1\}$*, this condition guarantees that* f_1 *is connected) and* $g_1 = f_1 + \sum_{i=k}^{n-1} x_i$ *or* $g_1 = f_1 + \sum_{i=0}^{k-1} x_i$*, then* (f_1, g_1) *is a connected complementary pair in Type-III. Also if* $f_2 = x_0x_{n-2} + x_1x_{n-2} + \sum_{i=2}^{n-3} x_i x_{n-1}$ *and* $g_2 = f_2 + x_0 + x_1 + x_{n-1}$ *or* $g_2 = f_2 + \sum_{i=2}^{n-2} x_i$*, then* (f_2, g_2) *is a connected complementary pair in Type-III. For* $n \leq 7$*,* (f_1, g_1) *and* (f_2, g_2) *cover all the connected complementary pairs up to re-labeling. Moreover, all the connected complementary pairs for* $n \leq 7$ *are recursively constructible from the primitive linear complementary pairs.*

[8] A connected graph is a graph where there is a path from every vertex to all other vertices. A connected quadratic Boolean function is a Boolean function whose corresponding graph is connected.

7.4 Conversions between Type-I/II/III complementary binary array pairs

Let (A_X, B_X) be a complementary binary array pair of size 2^n in Type-X, then by Theorem 29, $((N^2)^{\otimes n}A_X, (N^2)^{\otimes n}B_X)$ and $(N^{\otimes n}A_X, N^{\otimes n}B_X)$ are Type-Y and Type-Z pairs respectively, where X, Y and Z are taken from $\{I, II, III\}$. The converted pair (F_x, G_x) can be trivially transformed to a pair (F'_x, G'_x) that is defined over $\{0, 1, i, -1, -i\}$. The following three sections give a direct formula for the non-binary array pairs obtained from the conversions of the canonical binary pairs in Type-I and Type-II and the star graph pair in Type-III to Type-II/III, Type-I/III and Type-I/II respectively.

Let $V = (V_{00\ldots0}, V_{00\ldots1}, \ldots, V_{11\ldots1})$ be an array of size 2^n defined over $\{0, 1, i, -1, -i\}$, then V can be written as $M \cdot i^P = (M_{00\ldots0}i^{P_{00\ldots0}}, M_{00\ldots1}i^{P_{00\ldots1}}, \ldots, M_{11\ldots1}i^{P_{11\ldots1}})$, where M is a binary array of size 2^n defined by $M_k = |V_k|$, where $k \in F_2^n$ and P is an array of size 2^n defined over $\mathbb{Z}_4$ as

$$P_k = \begin{cases} 0 & \text{if } V_k = 1 \\ 1 & \text{if } V_k = i \\ 2 & \text{if } V_k = -1 \\ 3 & \text{if } V_k = -i \\ * & \text{if } V_k = 0 \end{cases}$$

where $k \in F_2^n$ and '*' can be any number. M can be considered as a truth table of a Boolean function defined over F_2. P can be considered as a truth table of a generalized Boolean function defined over $\mathbb{Z}_4$.

Theorem 1 tells us how to find the algebraic normal form for a Boolean function defined over F_2. So by applying Theorem 1, we find the algebraic normal form of M,

$m(x_0, \ldots, x_{n-1})$. The algebraic normal form for P is defined by,

$$p(x_0, x_1, \ldots, x_{n-1}) = \sum_{j=(j_0,\ldots,j_{n-1}) \in F_2^n} a_j x_0^{j_0} x_1^{j_1} \cdots x_{n-1}^{j_{n-1}} \pmod 4$$

where $a_j \in \mathbb{Z}_4$. Theorem 1 can be easily modified to find the algebraic normal form of Boolean functions defined over $\mathbb{Z}_4$. This modification can be stated as follows: *Let f be the truth table of an n-variable Boolean function defined over $\mathbb{Z}_4$ and C be as defined in section 2.3 (the coefficient vector in the ANF of f). Then*

$$C = fA_n$$

where

$$A_n = \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}^{\otimes n}$$

The above modification enables us to find the algebraic normal form $p(x_0, x_1, \ldots, x_{n-1})$ of the vector P. Thus, we can write the pair members as $m(x_0, \ldots, x_{n-1}) \cdot i^{p(x_0,\cdots,x_{n-1})}$, where $i = \sqrt{-1}$, $m(x_0, \ldots, x_{n-1})$ and $p(x_0, \cdots, x_{n-1})$ are the algebraic normal forms of M and P respectively.

The following two examples explain the conversions of the path graph array pair of size 2^3 (Type-I complementary array pair) to Type-II and Type-III respectively.

Example 1. Converting the path graph array pair of size 2^3, $(F = (-1)^{x_0x_1+x_1x_2}, G = (-1)^{x_0x_1+x_1x_2+x_2})$ to Type-II, gives us the following arrays $(N^2)^{\otimes n}F = (-1+i, 0, 1+i, 0, 0, 1-i, 0, 1+i) = \sqrt{2}\omega(i, 0, 1, 0, 0, -i, 0, 1) = \sqrt{2}\omega F'$ and $(N^2)^{\otimes n}G = (0, 1+i, 0, 1-i, 1+i, 0, -1+i, 0) = \sqrt{2}\omega(0, 1, 0, -i, 1, 0, i, 0) = \sqrt{2}\omega G'$, where $\omega = \frac{1+i}{\sqrt{2}}$. Since $(N^2)^{\otimes n}F = \sqrt{2}\omega F'$ and $(N^2)^{\otimes n}G = \sqrt{2}\omega G'$ form a complementary array pair in Type-II, then the pair (F', G') is also complementary in Type-II since $|F'_{k_0,k_1,k_2}|^2 + |G'_{k_0,k_1,k_2}|^2$ is constant for every $(k_0, k_1, k_2) \in F_2^3$ since $|\omega| = 1$. $F' = (i, 0, 1, 0, 0, -i, 0, 1)$ can be written as $M_{F'} \cdot i^{P_{F'}}$, where $M_{F'} = (1, 0, 0, 0, 1, 0, 1)$ and $P_{F'} = (1, *, 0, *, *, 3, *, 0)$. $G' = (0, 1, 0, -i, 1, 0, i, 1)$ can be written as $M_{G'} \cdot i^{P_{G'}}$, where $M_{G'} = (0, 1, 0, 1, 1, 0, 1, 1)$ and $P_{G'} = (*, 0, *, 3, 0, *, 1, *)$. Since

the '*' in $P_{F'}$ and $P_{G'}$ can be any number, we replace it by a certain number in $\mathbb{Z}_4$ so as to make $P_{F'}$ or $P_{G'}$ be as symmetric as possible so that the ANF of $P_{F'}$ and $P_{G'}$ can be in its simplest form . Thus, we change $P_{F'}$ from $(1,*,0,*,*,3,*,0)$ to $(1,3,0,0,1,3,0,0)$. We change $P_{G'}$ from $(*,0,*,3,0,*,1,*)$ to $(0,0,1,3,0,0,1,3)$. Now the ANFs of $M_{F'}$ and $M_{G'}$ equal x_0+x_2+1 and x_0+x_2 respectively. The ANF for $P_{F'}$ obtained by using Corollary 10 is $2x_2+3x_1+2x_1x_2+1$ which is a generalized Boolean function defined over $\mathbb{Z}_4$. The ANF for $P_{G'}$ obtained by using Corollary 10 is $x_1+2x_1x_2$ which is a generalized Boolean function defined over $\mathbb{Z}_4$. Therefore, the pair $((x_0+x_2+1)\cdot i^{2x_2+3x_1+2x_1x_2+1}, (x_0+x_2)\cdot i^{x_1+2x_1x_2})$ is a complementary array pair in Type-II.

Example 2. Converting the path graph array pair of size 2^3, $(F=(-1)^{x_0x_1+x_1x_2}, G=(-1)^{x_0x_1+x_1x_2+x_2})$ to Type-III, gives us the following arrays $N^{\otimes n}F=\frac{1}{\sqrt{2}}(1+i,1+i,-1+i,1-i,1+i,-1-i,1-i,1-i)=\omega(1,1,i,-i,i,-1,-i,-i)=\omega F'$ and $N^{\otimes n}G=\frac{1}{\sqrt{2}}(1+i,1+i,1-i,-1+i,-1-i,1+i,1-i,1-i)=\omega(1,1,-i,i,-1,1,-i,-i)=\omega G'$, where $\omega=\frac{1+i}{\sqrt{2}}$. Since $N^{\otimes n}F=\sqrt{2}\omega F'$ and $N^{\otimes n}G=\sqrt{2}\omega G'$ form a complementary array pair in Type-III, then the pair (F',G') is also complementary in Type-III since $|F'_{k_0,k_1,k_2}|^2+|G'_{k_0,k_1,k_2}|^2$ is constant for every $(k_0,k_1,k_2)\in F_2^3$ since $|\omega|=1$. $F'=(1,1,i,-i,i,-1,-i,-i)$ can be written as $M_{F'}\cdot i^{P_{F'}}$, where $M_{F'}=(1,1,1,1,1,1,1,1)$ and $P_{F'}=(0,0,1,3,1,2,3,3)$. $G'=(1,1,-i,i,-1,1,-i,-i)$ can be written as $M_{G'}\cdot i^{P_{G'}}$, where $M_{G'}=(1,1,1,1,1,1,1,1)$ and $P_{G'}=(0,0,3,1,2,0,3,3)$. Now the ANF of $M_{F'}$ and $M_{G'}$ equals the constant one for both of them. The ANF for $P_{F'}$ obtained by using Corollary 10 is $x_1+2x_1x_2+2x_0x_2+2x_0x_1$ which is a generalized Boolean function defined over $\mathbb{Z}_4$. The ANF for $P_{G'}$ obtained by using Corollary 10 is $3x_1+2x_1x_2+2x_0+2x_0x_2+2x_0x_1$ which is a generalized Boolean function defined over $\mathbb{Z}_4$. Therefore, the pair $(i^{x_1+2x_1x_2+2x_0x_2+2x_0x_1}, i^{3x_1+2x_1x_2+2x_0+2x_0x_2+2x_0x_1})$ is a complementary array pair in Type-III.

7.4.1 Converting Type-I to Type-II and Type-III

The only known Type-I complementary pair is the path graph pair. Converting the path graph pair, $(F_I, G_I) = ((-1)^f, (-1)^g)$, where $f = x_0x_1 + x_1x_2 + \cdots + x_{n-2}x_{n-1}$ and $g = f + x_{n-1}$ to Type-II and Type-III gives us the pairs $(F_{II}, G_{II}) = ((N^2)^{\otimes n}(-1)^f, (N^2)^{\otimes n}(-1)^g)$ and $(F_{III}, G_{III}) = (N^{\otimes n}(-1)^f, N^{\otimes n}(-1)^g)$ respectively. We transform (F_{II}, G_{II}) and (F_{III}, G_{III}) to pairs (F'_{II}, G'_{II}) and (F'_{III}, G'_{III}) respectively that are defined over $\{0, 1, i, -1, -i\}$. We then find the $m.i^p$ formulas for both (F'_{II}, G'_{II}) and (F'_{III}, G'_{III}).

Converting the path graph to Type-II

The function obtained by converting the first member of the path graph pair of size 2^n to Type-II (F'_{II}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \begin{cases} 1 & \text{if } n \textit{ is even} \\ 1 + \sum_{i=0}^{\frac{n-1}{2}} x_{2i} & \text{if } n \textit{ is odd} \end{cases}$$

$$p = \begin{cases} c + 3\sum_{i=0}^{n-1} x_i + 2\sum_{i=0}^{\frac{n-2}{2}} \sum_{j=i}^{\frac{n-2}{2}} x_{2i}x_{2j+1} & \text{if } n \textit{ is even} \\ c + 2\sum_{i=1}^{\frac{n-1}{2}} x_{2i} + 3\sum_{i=0}^{\frac{n-3}{2}} x_{2i+1} + 2\sum_{i=1}^{n-2} \sum_{j=\lceil \frac{i+1}{2} \rceil}^{\frac{n-1}{2}} x_i x_{2j} & \text{if } n \textit{ is odd} \end{cases}$$

where $c \in Z_4$.[9]

The function obtained by converting the second member of the path graph pair to Type-II (G'_{II}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \begin{cases} 1 & \text{if } n \textit{ is even} \\ \sum_{i=0}^{\frac{n-1}{2}} x_{2i} & \text{if } n \textit{ is odd} \end{cases}$$

$$p = \begin{cases} c + 3\sum_{i=0}^{\frac{n-2}{2}} x_{2i+1} + \sum_{i=0}^{\frac{n-2}{2}} x_{2i} + 2\sum_{i=0}^{\frac{n-2}{2}} \sum_{j=i}^{\frac{n-2}{2}} x_{2i}x_{2j+1} & \text{if } n \textit{ is even} \\ c + \sum_{i=0}^{\frac{n-3}{2}} x_{2i+1} + 2\sum_{i=1}^{n-2} \sum_{j=\lceil \frac{i+1}{2} \rceil}^{\frac{n-1}{2}} x_i x_{2j} & \text{if } n \textit{ is odd} \end{cases}$$

[9]Note that throughout this section, we do not need to know the value of the constant c in p as it has no effect since $|U \cdot mi^{c+p_1}| = |U \cdot mi^{p_1}|$, where $p = c + p_1$ and U is a unitary matrix of either Type-I, Type-II or Type-III.

where $c \in Z_4$.

Converting the path graph to Type-III

The function obtained by converting the first member of the path graph pair to Type-III (F'_{III}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \begin{cases} 1 & \text{if } n = 3k \\ 1 & \text{if } n = 3k+1 \\ 1 + \sum_{i=0, i\neq 3k+2}^{n-1} x_i & \text{if } n = 3k+2 \end{cases}$$

$$p = \begin{cases} c + \sum_{i=0}^{\frac{n-3}{3}} x_{3i+1} + 2\sum_{i=0, i\neq 3k+2}^{n-2} \sum_{j=i+1, j\neq 3k}^{n-1} x_i x_j & \text{if } n = 3k \\ c + 3\sum_{i=0}^{\frac{n-1}{3}} x_{3i} + 2\sum_{i=0, i\neq 3k+2}^{n-3} \sum_{j=i+1, j\neq 3k+1}^{n-1} x_i x_j & \text{if } n = 3k+1 \\ c + 3\sum_{i=0}^{\frac{n-2}{3}} x_{3i+1} + 2\sum_{i=1, i\neq 3k}^{n-3} \sum_{j=i+1, j\neq 3k+2}^{n-1} x_i x_j & \text{if } n = 3k+2 \end{cases}$$

where $c \in Z_4$.

The function obtained by converting the second member of the path graph pair to Type-III (G'_{III}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \begin{cases} 1 & \text{if } n = 3k \\ 1 & \text{if } n = 3k+1 \\ \sum_{i=0, i\neq 3k+2}^{n-1} x_i & \text{if } n = 3k+2 \end{cases}$$

$$p = \begin{cases} c + 3\sum_{i=0}^{\frac{n-3}{3}} x_{3i+1} + 2\sum_{i=0}^{\frac{n-3}{3}} x_{3i} + 2\sum_{i=0, i\neq 3k+2}^{n-2} \sum_{j=i+1, j\neq 3k}^{n-1} x_i x_j & \text{if } n = 3k \\ c + \sum_{i=0}^{\frac{n-1}{3}} x_{3i} + 2\sum_{i=0}^{\frac{n-4}{3}} x_{3i+1} + 2\sum_{i=0, i\neq 3k+2}^{n-3} \sum_{j=i+1, j\neq 3k+1}^{n-1} x_i x_j & \text{if } n = 3k+1 \\ c + \sum_{i=0}^{\frac{n-2}{3}} x_{3i+1} + 2\sum_{i=0}^{\frac{n-5}{3}} x_{3i+2} + 2\sum_{i=1, i\neq 3k}^{n-3} \sum_{j=i+1, j\neq 3k+2}^{n-1} x_i x_j & \text{if } n = 3k+2 \end{cases}$$

where $c \in Z_4$.

7.4.2 Conversion of Type-II to Type-I and Type-III

The only known Type-II complementary pair is the complete graph pair. Converting the complete graph pair, $(F_{II}, G_{II}) = ((-1)^f, (-1)^g)$, where $f = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} x_i x_j$ and $g = f + \sum_{i=0}^{n-1} x_i$ to Type-I and Type-III gives us the pairs $(F_I, G_I) = (N^{\otimes n}(-1)^f, N^{\otimes n}(-1)^g)$ and $(F_{III}, G_{III}) = ((N^2)^{\otimes n}(-1)^f, (N^2)^{\otimes n}(-1)^g)$ respectively. We transform (F_I, G_I) and (F_{III}, G_{III}) to pairs (F'_I, G'_I) and (F'_{III}, G'_{III}) respectively that are defined over $\{0, 1, i, -1, -i\}$. We then find the $m.i^p$ formulas for both (F'_I, G'_I) and (F'_{III}, G'_{III}).

Converting the complete graph to Type-I

The function obtained by converting the first member of the complete graph pair to Type-I (F'_I) can be written as $m \cdot i^p$ where $m = \sum_{(i_0,i_1,\ldots,i_{n-1}) \in F_2^n/(1,1,\ldots,1)} x_0^{i_0} x_1^{i_1} \ldots x_{n-1}^{i_{n-1}}$ and $p = c + 3x_{n-1}$ where $c \in F_2$. The function obtained by converting the second member of the complete graph pair to Type-I (G'_I) can be written as $m \cdot i^p$ where $m = \sum_{(i_0,i_1,\ldots,i_{n-1}) \in F_2^n/(1,1,\ldots,1)} x_0^{i_0} x_1^{i_1} \ldots x_{n-1}^{i_{n-1}}$ and $p = 3 + x_{n-1}$.

Converting the complete graph to Type-III

The function obtained by converting the first member of the complete graph pair to Type-III (F'_{III}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \begin{cases} 1 & \text{if } n \ is \ even \\ c + \sum_{i=0}^{n-1} x_i & \text{if } n \ is \ odd \end{cases}$$

where $c = 1$ if $n(mod\ 4) \equiv 1$ and equals to 0 otherwise.

$$p = \begin{cases} 0 & \text{if } n \ is \ odd \\ \sum_{i=0}^{n-1} x_i + 2 \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} x_i x_j & \text{if } n(mod\ 4) \equiv 0 \\ 1 + 3 \sum_{i=0}^{n-1} x_i + 2 \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} x_i x_j & \text{if } n(mod\ 4) \not\equiv 0 \end{cases}$$

The function obtained by converting the second member of the complete graph pair to Type-III (G'_{III}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \begin{cases} 1 & \text{if } n \ is \ even \\ c + \sum_{i=0}^{n-1} x_i & \text{if } n \ is \ odd \end{cases}$$

where $c = 0$ if $n(mod\ 4) \equiv 1$ and equals to 1 otherwise.

$$p = \begin{cases} 0 & \text{if } n \ is \ odd \\ 3\sum_{i=0}^{n-1} x_i + 2\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n(mod\ 4) \equiv 0 \\ 3 + \sum_{i=0}^{n-1} x_i + 2\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n(mod\ 4) \not\equiv 0 \end{cases}$$

7.4.3 Converting Type-III to Type-I and Type-II

Out of an infinite choice of pairs, we choose the star graph complementary pair in Type-III and convert it to a Type-I and Type-II complementary pair. Converting the star graph pair, $f = \sum_{i=0}^{n-1} x_i x_{n-1}$ and $g_1 = f + x_{n-1}$ or $g_2 = f + \sum_{i=0}^{n-2} x_i$ to Type-I and Type-II gives us the pairs $(F_I, G_{I_1} \text{ or } G_{I_2}) = ((N^2)^{\otimes n}(-1)^f, (N^2)^{\otimes n}(-1)^{g_1} \text{ or } (N^2)^{\otimes n}(-1)^{g_2})$ and $(F_{II}, G_{II_1} \text{ or } G_{II_2}) = (N^{\otimes n}(-1)^f, N^{\otimes n}(-1)^{g_1} \text{ or } N^{\otimes n}(-1)^{g_2})$ respectively. We transform $(F_I, G_{I_1} \text{ or } G_{I_2})$ and $(F_{II}, G_{II_1} \text{ or } G_{II_2})$ to pairs $(F'_I, G'_{I_1} \text{ or } G'_{I_2})$ and $(F'_{II}, G'_{II_1} \text{ or } G'_{II_2})$ respectively that are defined over $\{0, 1, i, -1, -i\}$. We then find the $m.i^p$ formulas for both $(F'_I, G'_{I_1} \text{ or } G'_{I_2})$ and $(F'_{II}, G'_{II_1} \text{ or } G'_{II_2})$.

Converting the star graph to Type-I

The function obtained by converting the first member of the star graph pair to Type-I (F'_I) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \sum_{(i_0, i_1, \ldots, i_{n-2}) \in F_2^{n-1}/(1,1,\ldots,1)} x_0^{i_0} x_1^{i_1} \ldots x_{n-2}^{i_{n-2}}$$

$$p = b_1 + 3x_{n-1} + b_2 x_{n-2} + 2x_{n-2}x_{n-1}$$

where $b_1 \in \mathbb{Z}_4$ and $b_2 = \begin{cases} (n-3)mod\ 4 & \text{if } n \ is \ even \\ (n-1)mod\ 4 & \text{if } n \ is \ odd \end{cases}$.

The function obtained by converting the second member (first option) of the star graph pair to Type-I (G'_{I_1}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \sum_{(i_0,i_1,\ldots,i_{n-2}) \in F_2^{n-1}/(1,1,\ldots,1)} x_0^{i_0} x_1^{i_1} \ldots x_{n-2}^{i_{n-2}}$$

$$p = b_1 + 3x_{n-1} + b_2 x_{n-2} + 2x_{n-2}x_{n-1}$$

where $b_1 \in \mathbb{Z}_4$ and $b_2 = \begin{cases} (n-1) mod\ 4 & \text{if } n\ is\ even \\ (n-3) mod\ 4 & \text{if } n\ is\ odd \end{cases}$.

The function obtained by converting the second member (second option) of the star graph pair to Type-I (G'_{I_2}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \sum_{(i_0,i_1,\ldots,i_{n-2}) \in F_2^{n-1}/(1,1,\ldots,1)} x_0^{i_0} x_1^{i_1} \ldots x_{n-2}^{i_{n-2}}$$

$$p = b_1 + x_{n-1} + b_2 x_{n-2} + 2x_{n-2}x_{n-1}$$

where $b_1 \in \mathbb{Z}_4$ and $b_2 = \begin{cases} (n-3) mod\ 4 & \text{if } n\ is\ even \\ (n-1) mod\ 4 & \text{if } n\ is\ odd \end{cases}$.

Converting the star graph to Type-II

The function obtained by converting the first member of the star graph pair to Type-II (F'_{II}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \begin{cases} 1 & \text{if } n\ is\ odd \\ b + \sum_{i=0}^{n-1} x_i & \text{if } n\ is\ even \end{cases}$$

where $b = \begin{cases} 0 & \text{if } n (mod\ 4) \equiv 0 \\ 1 & \text{if } n (mod\ 4) \not\equiv 0 \end{cases}$.

$$p = \begin{cases} b_1 + x_{n-1} + 2\sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n \ = \ 4k \\ b_2 + 3x_{n-1} + 2\sum_{i=0}^{n-2} x_i + 2\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n \ = \ 4k+1 \\ b_3 + 3x_{n-1} + 2\sum_{i=1}^{n-2} x_i + 2\sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n \ = \ 4k+2 \\ b_4 + x_{n-1} + 2\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n \ = \ 4k+3 \end{cases}$$

where b_1, b_2, b_3 and $b_4 \in \mathbb{Z}_4$ and k is a positive integer.

The function obtained by converting the second member (first option) of the star graph pair to Type-II (G'_{II_1}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \begin{cases} 1 & \text{if } n \ \textit{is} \ \textit{odd} \\ b + \sum_{i=0}^{n-1} x_i & \text{if } n \ \textit{is} \ \textit{even} \end{cases}$$

where $b = \begin{cases} 0 & \text{if } n(mod \ 4) \not\equiv 0 \\ 1 & \text{if } n(mod \ 4) \equiv 0 \end{cases}$.

$$p = \begin{cases} b_1 + 3x_{n-1} + 2\sum_{i=1}^{n-2} x_i + 2\sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n \ = \ 4k \\ b_2 + x_{n-1} + 2\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n \ = \ 4k+1 \\ b_3 + x_{n-1} + 2\sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n \ = \ 4k+2 \\ b_4 + 3x_{n-1} + 2\sum_{i=0}^{n-2} x_i + 2\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n \ = \ 4k+3 \end{cases}$$

where b_1, b_2, b_3 and $b_4 \in \mathbb{Z}_4$ and k is a positive integer.

The function obtained by converting the second member (second option) of the star graph pair to Type-II (G'_{II_2}) can be written as $m \cdot i^p$ where m and p are the following functions:

$$m = \begin{cases} 1 & \text{if } n \ \textit{is} \ \textit{odd} \\ b + \sum_{i=0}^{n-1} x_i & \text{if } n \ \textit{is} \ \textit{even} \end{cases}$$

where $b = \begin{cases} 0 & \text{if } n(mod \ 4) \not\equiv 0 \\ 1 & \text{if } n(mod \ 4) \equiv 0 \end{cases}$.

$$p = \begin{cases} b_1 + x_{n-1} + 2\sum_{i=1}^{n-2} x_i + 2\sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n = 4k \\ b_2 + 3x_{n-1} + 2\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n = 4k+1 \\ b_3 + 3x_{n-1} + 2\sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n = 4k+2 \\ b_4 + x_{n-1} + 2\sum_{i=0}^{n-2} x_i + 2\sum_{i=0}^{n-2}\sum_{j=i+1}^{n-1} x_i x_j & \text{if } n = 4k+3 \end{cases}$$

where b_1, b_2, b_3 and $b_4 \in \mathbb{Z}_4$ and k is a positive integer.

7.5 Construction of binary near-complementary pairs

In this section, we find near-complementary binary pairs with respect to Type-I and Type-II matrices simultaneously. For an array pair of size 2^n, (A, B), we define $P_{X,Y}(A, B) = \max(P_X(A, B), P_Y(A, B))$ where $P_X(A, B)$ and $P_Y(A, B)$ are the PAPRs with respect to Type-X and Type-Y matrices respectively, and $P_{X,Y}(A, B)$ is therefore, the maximum PAPR taken with respect to Type-X matrices and Type-Y matrices.

Type-I and Type-II near-complementary pairs

We see that the Type-I binary construction in 7.18 and the Type-II binary construction in 7.23 output the same pair when $c = \overleftarrow{c}$ and $d = \overleftarrow{d}$. Therefore, the following proposition follows.

Proposition 5. *If* $(A = (-1)^a, B = (-1)^b)$ *and* $(C = (-1)^c, D = (-1)^d)$ *are near-complementary binary pairs with respect to Type-I and Type-II unitary matrices, then* $(F = (-1)^f, G = (-1)^g)$*, where* f *and* g *are as follows*

$$\begin{aligned} f &= (a+b)(c+d) + a + d \\ g &= (a+b)(c+d) + b + c \end{aligned} \tag{7.29}$$

is a near-complementary pair with respect to Type-I and Type-II if $c = \overleftarrow{c}$ *and* $d = \overleftarrow{d}$*. We call this property of* (c, d)*, the reversal property. Moreover,* $P_{I,II}(F, G) = P_{I,II}(A, B)P_{I,II}(C, D)$.

The output pair of 7.29, (f, g), further satisfies $f = \overleftarrow{f}$ and $g = \overleftarrow{g}$ if $a = \overleftarrow{a}$ and $b = \overleftarrow{b}$. Thus by making use of 7.29, our problem of finding near-complementary pairs is restricted to finding near-complementary primitive pairs including those that satisfy the reversal property. We then use these near-complementary primitive pairs as initial pairs in 7.29 (note that the initial pair (c, d) should satisfy the reversal property). We performed a search on all quadratic functions (A more general search is desirable but computationally prohibitive. Moreover, our computational experiments suggest that quadratic functions give the pairs with lowest PAPRs). We searched all quadratics for a number of variables between 2 and 8 to find functions f, having $T = max(P_I((-1)^f), P_{II}((-1)^f))$ as small as possible. Assuming that we covered enough spectral points, we found that T is lower bounded by 4.0 when the number of variables is between 2 and 6, and lower bounded by 8.0 when the number of variables is 7 or 8. We then tried to find for each function another quadratic function so that together they form a near-complementary pair. We assumed that for a quadratic function f, the best pair is $(f, f + l)$ where l is a linear function (computational experiments suggested that this was a good idea). Assuming that we covered enough spectral points, we calculated $P_{I,II}(f, f + l)$ for every possible linear function. Table 7.1 shows the pairs found for a number of variables between 2 and 8 (For n = 7 and 8, we list only the functions that satisfy the reversal property). The table lists all the linear functions that form a pair with f, where f is an n-variable quadratic Boolean function with a low PAPR with respect to Type-I and Type-II matrices. The linear function $l_i = x_{i_0} + x_{i_1} + \ldots + x_{i_j}$ where $1 \leq i \leq 2^n - 1$ and $0 \leq i_j \leq 2^n - 1$ are the indices of the ones in the binary representation of integer i. For instance $l_5 = x_0 + x_2$ and $l_7 = x_0 + x_1 + x_2$ when $n = 3$. The table condenses the ANF of a Boolean function by replacing each x_i by i and each '+' by ','. For instance, $f = x_0x_2 + x_1x_3$ is condensed to $02, 13$.

The following two examples show two different ways of using 7.29.

Example 3. We found that $(A = (-1)^a, B = (-1)^b)$ is a primitive pair with $P_{I,II}(A, B) = 2$, where $a = 01$ and $b = a, 1$. Similarly, we found that $(C = (-1)^c, D = (-1)^d)$ is a

primitive pair satisfying the reversal property with $P_{I,II}(C, D) = 2$, where $c = 23, 24, 35, 45$ and $d = c, 2, 3, 4, 5$. Using 7.29, we get the following $(F = (-1)^f, G = (-1)^g)$ pair, where $f = 01, 23, 24, 14, 15, 35, 45, 4, 5$ and $g = 01, 23, 24, 14, 15, 35, 45, 1$. Then (F, G) is a pair with $P_{I,II}(F, G) = 4$ but (F, G) does not satisfy the reversal property so we can only use it recursively again as an (A, B) pair not as a (C, D) pair.

Example 4. We found that $(A = (-1)^a, B = (-1)^b)$ is a primitive pair with $P_{I,II}(A, B) = 2$, where $a = 01, 02, 13, 23$ and $b = a, 1$. Similarly, we found that $(C = (-1)^c, D = (-1)^d)$ is a primitive pair satisfying the reversal property with $P_{I,II}(C, D) = 2$, where $c = 45, 46, 57, 67$ and $d = c, 4, 5, 6, 7$. Using 7.29, we get the following $(F = (-1)^f, G = (-1)^g)$ pair, where $f = 02, 13, 23, 46, 06, 07, 16, 17, 26, 27, 36, 37, 01, 45, 57, 67, 6, 7$ and $g = f, 0, 1, 2, 3$. Then (F, G) is a pair with $P_{I,II}(F, G) = 4$ and since (A, B) satisfies the reversal property then so does (F, G). This gives us a pair that is recursively usable as either an (A, B) pair or a (C, D) pair.

n	f	l	$P_{I,II}(f, f+l)$
2	01	All linear functions	2.0
3	12	l_4, l_5, l_6, l_7	2.0
3	01,02,12	All linear functions except l_7	2.0
3	02,12	All linear functions except l_1	2.0
4	01,23	$l_7, l_{11}, l_{13}, l_{14}$	2.0
4	03,12,13,23	$l_6, l_8, l_{10}, l_{11}, l_{12}, l_{13}, l_{15}$	2.0
4*	01,02,13,23	l_3, l_5, l_{10}, l_{12}	2.0
4	02,03,12,13,23	$l_4, l_5, l_6, l_7, l_8, l_9, l_{10}$	2.0
4	02,13,23	$l_5, l_7, l_{10}, l_{11}, l_{12}, l_{13}, l_{14}, l_{15}$	2.0
5	02,04,13,14,24	$l_{14}, l_{15}, l_{23}, l_{26}, l_{27}$	2.0
5	02,03,13,14,24	l_{31}	2.25
5	02,03,04,13,14,24	$l_{10}, l_{15}, l_{21}, l_{30}$	2.0
5	02,03,04,13,14,24,34	$l_{10}, l_{14}, l_{20}, l_{23}, l_{27}, l_{28}, l_{31}$	2.0
5	02,03,04,14,23,24	l_{13}, l_{25}, l_{31}	2.0
5	02,03,04,13,14,23,24,34	$l_{15}, l_{21}, l_{22}, l_{27}$	2.0
6	02,04,05,13,14,15,24,35	All linear functions	4.0
6	02,04,05,13,14,15,24,35,45	l_{63}	2.0
6	02,04,05,13,14,24,25,35,45	All linear functions	4.0
6	02,03,05,13,14,15,24,25,35,45	All linear functions	4.0
6	02,03,04,05,13,14,15,24,25,35	All linear functions	4.0
6	02,03,04,05,13,14,15,24,25,34,35,45	$l_{21}, l_{22}, l_{41}, l_{42}$	2.0
6	02,03,04,05,14,15,23,24,25,35,45	l_{27}, l_{51}	2.0
7*	03,05,14,16,25,26,36,46	$l_{18}, \ldots, l_{123}$	4.0
7*	03,04,05,06,14,16,25,26,36,45,46,56	$l_{22}, \ldots, l_{125}$	4.0
7*	03,05,14,15,24,26,36,45,46,56	$l_{18}, \ldots, l_{127}$	4.0
7*	02,04,05,06,13,14,15,16,24,35,46,56	$l_1, \ldots, l_{127}$	4.0
7*	02,03,05,06,13,14,15,16,24,25,26,35,36,45,46,56	$l_4, \ldots, l_{123}$	4.0
7*	02,03,04,05,13,14,15,16,24,25,26,35,36,46	$l_1, \ldots, l_{127}$	4.0
8*	03,05,06,07,14,15,25,27,36,46,57,67	$l_{39}, \ldots, l_{255}$	4.0
8*	03,05,06,07,14,15,16,17,26,27,35,47,57,67	$l_{34}, \ldots, l_{255}$	4.0
8*	03,05,06,07,14,16,25,27,35,36,37,47,57,67	$l_{36}, \ldots, l_{239}$	4.0

Table 7.1: The '...' in $n = 7$ and $n = 8$ means that there are many other linear functions. The '*' means that the corresponding function, f, satisfies the reversal property, *i.e.* $f = \overleftarrow{f}$. The pair $(f, f + l)$ will satisfy the reversal property if l has an even number of terms.

Chapter 8

Conclusion

In the first half of the thesis, we studied the Walsh cryptographic criteria and the autocorrelation cryptographic criteria of Boolean functions. We also studied theoretical bounds on these criteria. We implemented a Boolean function database. The database holds information about cryptographic criteria of Boolean functions. It also calculates theoretical bounds for any valid combination of those cryptographic criteria.

In the second half of the thesis we studied the aperiodic autocorrelation spectrum of a Boolean function and some more spectral measures with respect to certain types of unitary matrix. We gave a survey on Golay complementary sequences and array pairs. The spectral property of Golay array pairs of size 2^n was re-expressed with reference to transforms formed from $2^n \times 2^n$ Type-I unitary matrices. We therefore called Golay complementary array pairs of size 2^n, "Type-I complementary" array pairs. This led us to define complementarity of array pairs of size 2^n with respect to two other transform types formed from $2^n \times 2^n$ unitary matrices called Type-II and Type-III matrices. Boolean function complementary pairs in Type-I, Type-II and Type-III were constructed. Non-binary alphabet complementary pairs in Type-I, Type-II and Type-III were constructed by converting binary pairs in Type-I, Type-II and Type-III to Type-II/III, Type-I/III and Type-I/II respectively. Boolean functions which are near-complementary pairs with respect to Type-I and Type-II were also constructed.

This thesis proposes the following future work,

1. There is still much work to be done to improve the Boolean function database. For

instance, the algebraic immunity criterion should be implemented. Constructions of good cryptographic functions should also be added to the website.

2. The Boolean function pairs constructed in Type-I, Type-II and Type-III are either affine or quadratic functions. This raises the question about the existence of binary pairs of higher algebraic degree in Type-I, Type-II and Type-III. So we propose answering this question as a future work.

3. Cryptanalysts exploit the weakness in the criteria related to the Walsh spectrum and the periodic autocorrelation spectrum of Boolean functions to break cryptosystems based on Boolean functions. However, the use of the aperiodic autocorrelation and some other spectral measures in cryptanalysis have been studied in [25, 9]. Reference [9] gave cryptographic criteria that are related to the aperiodic autocorrelation spectrum. In [25] the definition of nonlinearity has been generalized to measure the distance of Boolean functions to affine functions over higher alphabets rather than just the binary alphabet and so the nonlinearity is calculated with respect to more general spectra rather than just the Walsh transform spectrum. In other words, the definition of nonlinearity has been generalized to the PAPR with respect to certain types of unitary matrix. These studies suggest that cryptosystems based on Boolean functions can be attacked by exploiting the weaknesses in their generalized nonlinearity. Therefore, the constructions of cryptographic Boolean functions should be expanded to include Boolean functions with higher PAPR with respect to certain types of unitary matrix. So we propose constructing good cryptographic Boolean functions having higher PAPR with respect to certain types of unitary matrix as a future work. We also propose implementing the aperiodic autocorrelation criteria defined in [9] and the generalized nonlinearity criterion in the Boolean function database.

References

[1] An Braeken. Cryptographic properties of Boolean functions and S-Boxes, *Phd Thesis, March 2006*, http://www.cosic.esat.kuleuven.be/publications/thesis-129.pdf, Last visited 2008.

[2] E. Biham and A. Shamir. (1990). Differential Cryptanalysis of DES-like Cryptosystems. *Advances in Cryptology CRYPTO '90*. Springer-Verlag. 221.

[3] P.B. Borwein and R.A. Ferguson, A complete description of Golay pairs for lengths up to 100, *Mathematics of Computation* 73 (2003), 967985.

[4] C. Carlet, Boolean functions for cryptography and error correcting codes. Preprint, 2007.

[5] C. Carlet, On the coset weight divisibility and nonlinearity of resilient and correlation-immune functions. *In Proceedings of the 2nd international conference (SETA'01), Discrete Mathematics and Theoretical Computer Science(1999)*, pp. 131-144.

[6] Carlet, C., and Guillot, P. A new representation of Boolean functions. *In Proceedings of AAECC13 (1999), no. 1719 in Lecture Notes in Computer Science, Springer-Verlag.*

[7] N. Courtois and W. Meier, Algebraic Attacks on Stream Ciphers with Linear Feedback. *Advances in cryptologyEUROCRYPT 2003, Lecture Notes in Computer Science* 2656, pp. 346-359, Springer, 2002.

[8] J.A. Davis and J. Jedwab, Peak-to-mean power control in OFDM, Golay complementary sequences, and Reed-Muller codes, *IEEE Trans. Information Theory* vol. 45 pp. 23972417, 1999.

[9] Lars Eirik Danielsen, T. Aaron Gulliver and Matthew G. Parker Aperiodic Propagation Criteria for Boolean Functions, *Inform. Comput.*, 204, 5, pp. 741-770, 2006.

[10] M. Dymond. Barker arrays: existence, generalization and alternatives. PhD thesis, University of London, 1992.

[11] S. Eliahou, M. Kervaire, and B. Saffari. A new restriction on the lengths of Golay complementary sequences. *J. Combin. Theory (A)*, 55:4959, 1990.

[12] F. Fiedler, J. Jedwab and M.G. Parker, A multi-dimensional approach to the construction and enumeration of Golay complementary sequences, *J. Combinatorial Theory (Series A)*, 2007.

[13] M.J.E. Golay. Static multislit spectrometry and its application to the panoramic display of infrared spectra. *J. Opt. Soc. Amer.*, 41:468472, 1951.

[14] M.J.E. Golay. Complementary series. *IRE Trans. Inform. Theory*, IT-7:8287, 1961.

[15] M.J.E. Golay. Note on Complementary series, *Proc. IRE* 50:84, 1962.

[16] J. Jedwab and M.G. Parker, Golay complementary array pairs, *Designs, Codes and Cryptography*, vol. 44 pp. 209216, 2007.

[17] J. Jedwab, What can be used instead of a Barker sequence ?, Contemporary Math., 2008.

[18] S. Kavut and M. Yucel, Generalized Rotation Symmetric and Dihedral Symmetric Boolean Functions - 9 variable Boolean Functions with Nonlinearity 242. *Cryptology ePrint Archive*, Report 2007/308, August 8, 2007, http://eprint.iacr.org/.

[19] H.D. Luke. Sets of one and higher dimensional Welti codes and complementary codes. *IEEE Trans. Aerospace Electron. Systems*, AES-21:170179, 1985.

[20] S. Maitra, Balanced Boolean Function on 13-variables having Nonlinearity strictly greater than the Bent Concatenation Bound, *Cryptology ePrint archive, http://eprint.iacr.org*, 2007/309.

[21] S. Maitra, On nonlinearity and Autocorrelation properties of Corelation Immune Boolean Functions, *Journal of Information Science and Engineering* 20,3305-323 (2004)

[22] M. Matsui, Linear cryptanalysis method for DES cipher. *Advances in Cryptology - EUROCRYPT93, no. 765 in Lecture Notes in Computer Science. Springer-Verlag*, pp. 386-397, 1994.

[23] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, Handbook of applied cryptography, *CRC Press, 1996, online-version: http://www.cacr.math.uwaterloo.ca/hac/.*

[24] N. Ohyama, T. Honda, and J. Tsujiuchi. An advanced coded imaging without side lobes. *Optics Comm.*, 27:339344, 1978.

[25] M.G.Parker, Generalised S-Box Nonlinearity, *NESSIE Public Document*, 11.02.03 NES/DOC/UIB/WP5/020/A

[26] M.G.Parker,K.G.Paterson and C.Tellambura, Golay Complementary Sequences, *Wiley Encyclopedia of Telecommunications*, Editor: J.G.Proakis, Wiley Interscience,2002.

[27] M.G.Parker and C.Tellambura, Generalised Rudin-Shapiro Constructions WCC2001 International Workshop on Coding and Cryptography, Paris(France), Jan 8-12, 2001. Electronic Notes in Discrete Mathematics, 6, April 2001

[28] E. Pasalic and T. Johansson, Further results on the relation between nonlinearity and resiliency of Boolean functions. *In IMA Conference on Cryptography and Coding,number 1746 in Lecture Notes in Computer Science*, pages 35-45. Springer-Verlag, 1999.

[29] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts and J.Vandevalle. Propagation characteristics of Boolean functions, *Advances in Cryptology, EUROCRYPT'90, Lecture Notes in Computer Sciences*, Springer Verlag no. 473, pp. 161-173, 1991.

[30] O. S. Rothaus. On bent functions. *J. Comb. Theory*, 20A, pp. 300-305, 1976.

[31] W. Rudin, Some theorems on Fourier coefficients, *Proc. Amer. Math. Soc.* 10, pp. 855859, 1959.

[32] I. Semaev. Lecture Notes on Advanced Cryptology,*INF247 course at the department of Informatics in the university of Bergen*, Spring 2007.

[33] P. Sarkar and S. Maitra, Nonlinearity bounds and constuction of resilient Boolean functions. *In Advances in Cryptology - Crypto 2000*, pages 515-532, Berlin, 2000. Springer-Verlag. Lecture Notes in Computer Science Volume 1880.

[34] P. Sarkar. Spectral domain analysis of correlation immune and resilient Boolean functions. *Cryptology ePrint archive (http://eprint.iacr.org)*. Reprint 2000/049, September 2000.

[35] P. Sarkar. A note on the spectral characterization of correlation immune Boolean functions. *Information Processing Letters*, 74(5-6),pp. 191-195. 2000.

[36] K.-U. Schmidt. On Cosets of the Generalized First-Order Reed-Muller Code with Low PMEPR. *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 3220-3232, July 2006.

[37] J. Seberry. X. M. Zhang, and Y. Zheng. Nonlinearly balanced boolean functions and their propagation characteristics. *In Advances in Cryptology - CRYPTO'93, volume 773, Lecture Notes in Computer Science*, pages 49-60. Springer-Verlag, Berlin, Heidelberg, New York, 1994. http://citeseer.ist.psu.edu/seberry93nonlinearly.html

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

[39] T. Siegenthaler, Correlation-Immunity of nonlinear combining functions for cryptograhic applications. *IEEE Transcations on Information Theory*, vol. IT-30, No 5, pp. 776-780, 1984.

[40] T. Siegenthaler, Decrypting a class of stream ciphers using ciphertext only. *IEEE Transactions on Computer, vol. C-34*, No 1, pp. 81-85, 1985.

[41] R.J. Turyn. Hadamard matrices, Baumert-Hall units, four-symbol sequences, pulse compression, and surface wave encodings. *J. Combin. Theory (A)*, 16:313333, 1974.

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

[43] Xiao Guo-Zhen, C. Ding and W. Shan. The stability theory of stream ciphers, *vol. LNCS 561, Springer Verlag*, 1991.

[44] Yu. Tarannikov, On resilient Boolean functions with maximal possible nonlinearity, *Proceedings of Indocrypt 2000, Lecture Notes in Computer Science*, V. 1977, pp. 19-30, Springer-Verlag, 2000.

[45] A.F. Webster and S.E. Tavares. On the design of S-boxes. *In Advances in Cryptology - CRYPTO'85, no. 219 in Lecture Notes in Computer Science*, pp. 523-534. Springer-Verlag, 1985.

[46] X.-M. Zhang and Y. Zheng, GAC - The criterion for global avalanche characteristics of cryptographic functions. *Journal of Universal Computer Science*, 1(5), pp. 320-337, 1995.

[47] Zhang, X. M., and Zheng, Y. Auto-correlations and new bounds on the nonlinearity of boolean functions. *In Advances in Cryptology - EUROCRYPT'96 (1996), vol. 1070, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, Heidelberg, New York, pp. 294–306. http://citeseer.ist.psu.edu/zhang96autocorrelations.html

[48] Y. Zheng and X.-M. Zhang. Improving upper bound on the nonlinearity of high order correlation immune functions. *Proceedings of Selected Areas in Cryptography 2000, Lecture Notes in Computer Science 2012*, pp. 262-274, 2001.