UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

DOKTORSKA DISERTACIJA
(DOCTORAL THESIS)

KARAKTERIZACIJA POSPLOŠNIH ZLOMLJENIH FUNKCIJ IN NEKATERE DRUGE KRIPTOGRAFSKE TEME

(CHARACTERISATION OF GENERALIZED BENT FUNCTIONS AND SOME OTHER TOPICS RELATED TO CRYPTOGRAPHY)

SAMIR HODŽIĆ

KOPER, 2017

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

DOKTORSKA DISERTACIJA
(DOCTORAL THESIS)

KARAKTERIZACIJA POSPLOŠNIH ZLOMLJENIH FUNKCIJ IN NEKATERE DRUGE KRIPTOGRAFSKE TEME

(CHARACTERISATION OF GENERALIZED BENT FUNCTIONS AND SOME OTHER TOPICS RELATED TO CRYPTOGRAPHY)

SAMIR HODŽIĆ

KOPER, 2017

MENTOR: IZR. PROF. DR. ENES PASALIC
SOMENTOR: DOC. DR. MARKO OREL

Acknowledgement

I owe my deepest gratitude to my supervisor Professor Enes Pasalic for the continuous support of my PhD study and research, enthusiasm and encouragement. Without his patience and sacrifice, this thesis would not have been possible. I would also like to thank to my co-supervisor Marko Orel for his support, availability and constructive suggestions.

I would like to thank the Department of Mathematics at University of Primorska, for giving me the opportunity to do research studies.

Posebnu zahvalnost upućujem mojim roditeljima, bratu, iskrenim prijateljima, kao i mojoj ljepšoj polovini na vječitoj podršci i razumjevanju mojih ciljeva. Njihova podrška je uvijek bila moja snaga i ostat će moja inspiracija kroz život.

Samir Hodžić

This work is supported part by "Agencija za raziskovalno dejavnost Republike Slovenije", research program and "Mladi raziskovalec" research program.

Abstract

CHARACTERISATION OF GENERALIZED BENT FUNCTIONS AND SOME OTHER TOPICS RELATED TO CRYPTOGRAPHY

This thesis considers three separate topics, all related to symmetric-key cryptography from both the design and the security perspective.

The first topic is the study of generalized bent (gbent) functions. Bent functions, or maximally nonlinear Boolean functions, have attracted intense research interest since their introduction by O.S. Rothaus in 1976, however a complete characterization remains elusive. Generalized bent functions widen the codomain of these functions from $\mathbb{Z}_2$ to $\mathbb{Z}_q$ ($q \geq 2$ any positive integer), and they are of interest because of their applications in the design of OFDM and MC-CDMA communication systems. In this thesis, we deduce a complete characterisation of gbent functions when q is a power of 2, which is the most interesting case due to applications. Depending on the parity of n, we show that a gbent function is a $(k-1)$-dimensional affine space of bent functions or semi-bent functions with certain interesting additional properties. In addition, we provide the first generic construction methods of gbent functions for n even or odd, and for any even q.

The second topic we consider is the analysis of stream and block ciphers from the design and cryptoanalysis point of view. As a typical representative of a hardware oriented design in stream ciphers, a nonlinear filter generator consists of a single linear feedback shift register (LFSR) and a nonlinear (vectorial) Boolean function F which processes a fixed subset of n stages of the LFSR (usually called the taps). Among various cryptanalytic approaches which utilize weaknesses of the filtering function, the guess and determine cryptanalysis is a powerful cryptanalytic tool for these schemes which does not depend on the filtering function, but rather on the selection of size of LFSR, the primitive polynomial used and the tapping sequence (tap positions used to provide F with the inputs). The important issue of finding (sub)optimal solutions for selecting tap positions is comprehensively treated in the dissertation. Two algorithms for the purpose of selecting taps (sub)optimally are presented, where we show that the selections of tap positions in real-life stream ciphers such as SOBER-t32, SFINKS and Grain-128 could have been (slightly) further optimized with respect to guess and determine cryptanalysis. In connection to nonlinear filter generators, the two well-known generic cryptanalytic methods which utilize certain algebraic properties of the function F in order to break the cipher, are known as Algebraic attacks (AA) and Fast algebraic attacks (FAA). However, the computational complexity of estimating the resistance of F to this type of cryptanalysis becomes large for $n \geq 30$. Therefore, in the dissertation we propose an efficient probabilistic algorithm (with high success rate) for determining the resistance of

a random Boolean function against AA and FAA. The algorithm employs partial linear relations, derived from the decomposition of an arbitrary nonlinear Boolean function into many small partial linear subfunctions by using disjoint sets of input variables.

As our final topic, we consider polynomials without linear structures. While the resistance of block ciphers to differential cryptanalysis relies heavily on the differential properties of vectorial Boolean functions (represented as polynomials), in order to achieve a high security level it is necessary that these contain no linear structures. In the dissertation we identify several new infinite classes of polynomials which cannot possess linear structures. While the linear structures of monomials and binomials are quite easy to handle, the existence of linear structures for arbitrary polynomials over finite fields is harder to analyze. Nevertheless, we provide a few interesting results in this direction, including some particular cases when these polynomials contain an arbitrary number of terms.

Math. Subj. Class (2010): 94A60, 11T71

Key words: Generalized bent functions, $\mathbb{Z}_q$-bent functions, Gray maps, (Relative) Difference sets, (Generalized) Marioana-McFarland class, Stream ciphers, Filtering generator, Guess and determine cryptanalysis, Tap positions, (Fast) Algebraic attacks, Algebraic immunity, Derivatives, Linear structures, Planar mappings.

Izvleček

KARAKTERIZACIJA POSPLOŠNIH ZLOMLJENIH FUNKCIJ IN NEKATERE DRUGE KRIPTOGRAFSKE TEME

Disertacija preučuje tri ločene teme, ki so povezane s kriptografijo simetričnih ključev tako z vidika dizajna kot tudi z vidika zaščite.

Prva tema preučuje posplošene zlomljene funkcije. Zlomljene funkcije oz. maksimalno nelinearne Boolove funkcije so podvržene raziskovanju že od leta 1976, ko jih je vpeljal O.S. Rothaus. Kljub temu se njihova popolna karakterizacija zdi nemogoča. Posplošene zlomljene funkcije, ki imajo kodomeno $\mathbb{Z}_2$ zamenjano s kolobarjem $\mathbb{Z}_q$ ($q \geq 2$ je celo število), so zanimive zaradi uporabe pri konstrukciji komunikacijskih sistemov OFDM in MC-CDMA. Disertacija vsebuje popolno karakterizacijo posplošenih zlomljenih funkcij, če je q potenca števila 2, kar predstavlja najzanimivejši primer z vidika uporabe. Pokazali bomo, da je, v odvisnosti od tega, ali je n sod oz. lih, posplošena zlomljena funkcija enaka $(k-1)$-razsežnemu afinemu prostoru zlomljenih oz. semi-zlomljenih funkcij, ki imajo nekatere dodatne lastnosti. Predstavili bomo tudi prvo generično konstrukcijsko metodo za posplošene zlomljene funkcije za sode in lihe n in za poljuben sod q.

Druga tema preučuje analizo tokovnih in bločnih šifer z vidika konstrukcije in kriptoanalize. Nelinearen filtrirni generator, ki je pomemben pri konstrukcijah tokovnih šifer, je sestavljen iz linearnega pomičnega registra LFSR in nelinearne (vektorske) Boolove funkcije F, ki obdeluje podmnožico n fiksnih celic registra LSFR. Med številnimi kriptoanalitičnih pristopi je še posebej pomembna kriptoanaliza tipa ugani-in-določi. Slednja je namreč neodvisna od filtrirne funkcije in sloni na izbiri velikosti registra LFSR, primitivnega polinoma in zaporedja fiksnih celic. Zajeten del disertacije je namenjen (sub)optimalni izbiri fiksnih celic. Predstavljena sta dva algoritma za izbiro teh celic, med drugim pa pokažemo tudi, da je mogoče izbor fiksnih celic pri nekaterih tokovnih šifrah iz vsakdanjega življenja (SOBER-t32, SFINKS, Grain-128) še nekoliko izboljšati z vidika kriptoanalize tipa ugani-in-določi. V povezavi s filtrirnimi generatorji so med generičnimi kriptoanalitičnimi metodami, ki izkoriščajo določene algebraične lastnosti funkcije F, znani predvsem algebraični napadi (AA) in hitri algebraični napadi (FAA). Računska kompleksnost za oceno zaščite funkcije F proti tovrstnim napadom je zelo velika za $n \geq 30$. V disertaciji predstavimo učinkovit verjetnostni algoritem za določanje zaščite slučajne Boolove funkcije proti napadom AA in FAA. Algoritem bazira na delnih linearnih relacijah, ki jih dobimo pri dekompoziciji poljubne nelinearne Boolove funkcije na več majhnih delnih linearnih podfunkcij z uporabo disjunktnih množic vhodnih spremenljivk.

V zadnjem delu disertacije preučujemo polinome, ki nimajo linearnih struktur. Zaščita bločnih šifer pri diferenčni kriptoanalizi sloni na diferenčnih lastnostih vek-

torskih Boolovih funkcij, ki so v obliki polinomov. Za dobro zaščito je pomembno, da le-ti nimajo linearnih struktur. V disertaciji je predstavljenih več novih neskončnih razredov polinomov, ki nimajo linearnih struktur. Medtem ko je preučevanje linearnih struktur pri monomih in binomih nad končnim obsegom relativno enostavno, je slednje pri splošnih polinomih precej težje. Kljub temu je v disertaciji predstavljenih tudi nekaj rezultatov iz tega področja.

Math. Subj. Class (2010): 94A60, 11T71

Key words: Posplošene zlomljene funkcije, $\mathbb{Z}_q$-zlomljene funkcije, Grayeve preslikave, (Relativne) diferenčne množice, (Posplošeni) Marioana-McFarlandov razred, Tokovne šifre, Filtrirni generator, Ugani-in-določi kriptoanaliza, Pozicije fiksnih celic, (Hitri) algebraični napadi, Algebraična imunost, Odvodi, Linearne strukture, Ravninske preslikave.

Contents

Index of Figures **xi**

Index of Tables **xiii**

1 Introduction **1**

2 Boolean functions and filtering generators **11**
2.1 Boolean functions 11
2.2 Generalized Boolean functions 15
2.3 Sylvester-Hadamard matrix 16
2.4 Nonlinear filtering generator 17
2.4.1 Overview of FSGA and GFSGA 17

3 Generalised bent (gbent) functions **21**
3.1 Motivation and Conjecture on GWHT 23
3.1.1 New GWHT formula 26
3.2 Sufficient conditions for gbent property (q even) 30
3.2.1 Equivalent forms of conditions ($\triangle$) and ($\square$) 33
3.2.2 Necessary and sufficient conditions for the GMMF class 35
3.2.3 Fulfilling the necessary conditions for gbent property 36
3.3 Full characterization of generalized bent functions 38
3.3.1 On the Sylvester-Hadamard matrix 38
3.3.2 Necessary and sufficient conditions ($q = 2^k$) 40
3.3.3 Gbent conditions in terms of affine (semi-)bent spaces 43
3.3.4 Equivalence of gbent functions 45
3.3.5 $\mathbb{Z}_q$-bent functions and relative difference sets 48
3.3.6 The dual and Gray map of gbent functions 50
3.3.7 The dual of a gbent function 50
3.3.8 The Gray map of gbent functions 51
3.4 Construction methods for generalized bent functions 53
3.4.1 Problem description 53
3.4.2 Construction of gbent functions using MM class 55
3.4.3 Disjoint spectra semi-bent functions in the MM class 55
3.4.4 Non-trivial selection of component functions, n odd 57
3.4.5 The construction when n is even 61
3.4.6 Illustrating the construction details - an example 61

3.5 Generalized bent functions constructed out of two (generalised) Boolean functions 62
3.5.1 Generalized bent functions from two bent Boolean functions . . . 64
3.5.2 Generalized bent functions using direct sum of gbent functions . 66
3.5.3 Generalized bent functions from one bent and one gbent function 67
3.5.4 Concatenation of generalized bent Boolean functions 68
3.5.5 Generalized bent functions defined on $\mathbb{Z}_2^n$, n odd 68
3.5.6 Generalized bent functions defined on $\mathbb{Z}_2^n$, n even 71
3.5.7 Construction methods for generalized bent functions in $\mathcal{GB}_n^{4s}$ . . 72

4 Optimizing the placement of tap positions 77
4.1 Complexity versus the number of repeated equations 79
4.2 Two algorithms towards an optimal selection of taps 83
4.3 GFSGA with a variable sampling step 88
4.3.1 The number of repeated equations for $GFSGA^*$ 90
4.3.2 Two specific modes of $GFSGA^*$ 92
4.3.3 $GFSGA^*_{(1)}$ mode of attack 93
4.3.4 $GFSGA^*_{(2)}$ mode of attack 94
4.4 Comparision between $GFSGA$, $GFSGA^*_{(1)}$ and $GFSGA^*_{(2)}$ 95
4.4.1 Overview of the algorithms for tap selection 95
4.4.2 Full positive difference sets versus algorithmic choice 96
4.4.3 Further examples and comparisons 98
4.5 Employing GFSGA in other settings 100
4.5.1 GFSGA applied to single-output nonlinear filter generators . . . 100
4.5.2 Applying GFSGA to NFSR-based ciphers 102
4.5.3 Grain-128 tap selection 105

5 Estimating the algebraic properties of Boolean functions for large n 107
5.1 A probabilistic decomposition algorithm for nonlinear Boolean functions 108
5.2 Estimating the resistance against AA and FAA 115
5.2.1 Resistance to AA 115
5.2.2 Resistance to FAA 117
5.2.3 An algorithm for estimating the resistance against AA and FAA 119

6 On derivatives of polynomials over finite fields through integration 127
6.1 Linear structures and derivatives 128
6.1.1 Some preliminary results using integration formula 130
6.1.2 Linear structures of mappings over finite fields and Boolean functions 131
6.2 Upper bounds on degree of planar mappings 134

7 Conclusions 139

Bibliography 141

Index 149

8 Povzetek v slovenskem jeziku **151**
Kazalo 160
Stvarno kazalo 163

List of Figures

1.1 Scheme of a classic cryptosystem 2
1.2 Symmetric-key encryption schemes 2
1.3 Filtering generator with tap positions 5

8.1 Shema klasičnega kriptosistema 152
8.2 Sheme pri enkripciji simetričnih ključev 152
8.3 Filtrirni generator 154

List of Tables

3.1 Vectors $W(u)$ for all $u \in \mathbb{Z}_2^5$. 62

4.1 The LFSR state bits at given tap positions for $\sigma = 2$. 81
4.2 The scheme of all possible differences for the set D. 81
4.3 The scheme of all differences for $D = \{3, 4, 1, 2\}$. 82
4.4 Specifications of difference sets for LFSRs of different lengths. 86
4.5 Time complexities for finding tap positions in Table 5. 86
4.6 The scheme of all differences for $D = \{2, 5, 4, 2\}$. 91
4.7 Complexity comparision of all three GFSGA modes for "bad" tap choices. 97
4.8 Complexity comparison of GFSGA modes - algorithmic selection of taps. 97
4.9 Complexity comparision - full positive difference sets versus algorithmic choice. 98
4.10 Repeated bits attained by sampling steps σ_i defined by (4.10). . . . 99
4.11 Repeated bits attained by sampling steps σ_i defined by (4.11). . . . 100
4.12 Recovered bits obtained by sampling step $\sigma_i = 1$ 104
4.13 Repeated bits attained by sampling step $\sigma_i = 1$ 105
4.14 Time complexity of different modes of GFSGA on LFSR of Grain-128 106
4.15 Time complexity of different modes of GFSGA on NFSR of Grain-128 106

5.1 The time complexity of our algorithm versus previous works. 120
5.2 A time complexity comparison for $30 \leq n \leq 40$. 121
5.3 Estimation the upper bound on the AI values of functions in [124]. 123
5.4 Estimation the upper bound on the resistance of functions in [124] against FAA. 123

Chapter 1

Introduction

Since the beginning of written history there have been attempts to keep physically recorded information confidential. Society continues to demand methods for securing sensitive information, however due to humankind's relatively recent leap into the information age the alphabet has been reduced to the 0s and 1s of electronic data, and thus the process of encoding has become ever more mathematical. The techniques used to protect data belong to the field of cryptography; the science of information and communication security.

The fundamental objective of cryptography is to enable two persons to communicate over an insecure channel in such a way that an adversary (a third party) is unable to recover their message (called the plaintext) from what is sent in its place over the channel (the ciphertext). More generally, it is about constructing and analyzing systems (protocols) which prevent third parties from reading private messages. On the other hand, cryptanalysis deals with breaking such systems. In general, cryptology is the all-inclusive term for the study of communication over insecure channels, and it encompasses the interrelated areas of cryptography and cryptanalysis. Modern cryptography exists at the intersection of various disciplines of mathematics, computer science, and electrical engineering.

Applications of cryptography are present in many aspects of our society, and they include authentication and encryption (bank cards, wireless telephone, e-commerce), access control (car lock systems, ski lifts) and payment (prepaid telephone cards, e-cash). Behind the all previously mentioned applications, an underlying cryptographic system has to satisfy a number of security goals. Some important aspects in information security are data confidentiality, data integrity, authentication, and non-repudiation, and some of these goals will be elaborated later in the framework of Boolean functions.

A classic example of a cryptosystem is depicted in Figure 8.1. Such a cryptosystem primitive is also called symmetric-key encryption algorithm, since the transmitted message (plaintext) is encrypted (into ciphertext) and decrypted with the same secret key which is shared between both sender and recipient. Symmetric-key cryptography comprises two large families of cryptographic primitives, namely block and stream ciphers (see Figure 8.2). Since both block and stream ciphers provide significant performance improvement compared to public key encryption techniques, they are commonly used as encryption schemes in practice. However, the design rules for

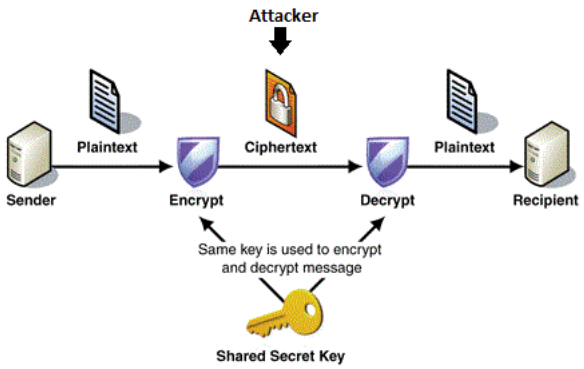

Figure 1.1: Scheme of a classic cryptosystem

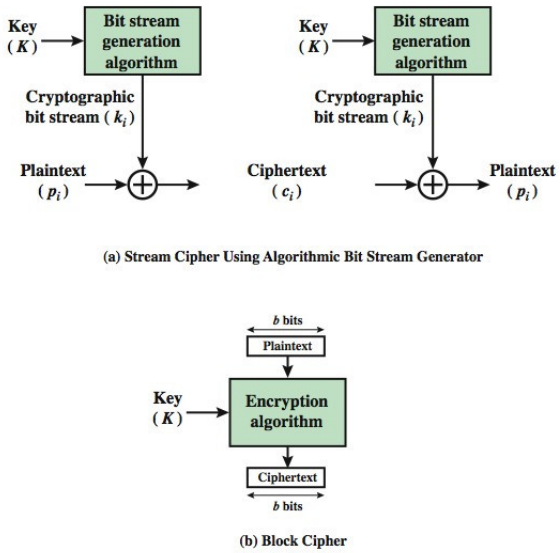

Figure 1.2: Symmetric-key encryption schemes

these two primitives are quite different.

The basic underlying idea in the design of block ciphers is to divide the plaintext into blocks (the length is always a power of two, usually 64,128 or 256 bits), and encode each block separately. The design of encryption algorithm that corresponds to a block cipher (see Figure 8.2) uses certain cryptographic primitives known as *substitution boxes* (S-boxes, or vectorial Boolean functions), which essentially can be viewed as a collection of Boolean functions whose selection and cryptographic properties are application/design dependant. The encryption of each plaintext block passes through multiple applications of the same S-box layer of the block cipher, which stands for the concept of confusion (each bit of the ciphertext should depend in a very complicated manner on plaintext and secret key bits). In addition each encryption round employs a linear layer, where also the so-called round (se-

cret) key is added, which then corresponds to the concept of diffusion (which can be roughly considered as the property that the intermediate ciphertext bits, after applying one encryption round, depend on many input bits). The concepts of confusion and diffusion were introduced by Claude E. Shannon in his classified report *A Mathematical Theory of Cryptography* [105] in 1945. Although good confusion and diffusion properties are relatively easily achieved, due to iterative process of processing the same plaintext block several times (typically 10-30 times), well designed stream ciphers are commonly slightly faster than block ciphers. Some of the well-know and prominent block ciphers today, based on the use of either Feistel or SP (Substitution Permutation) networks, are Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), Triple DES, Twofish, Serpent and Advanced Encryption Standard (AES) (the current encryption standard).

On the other hand, stream ciphers commonly employ (non)linear shift registers which roughly correspond to finite state machines capable of storing and manipulating its state bits in (non)linear manner. Most commonly, a portion of these state bits is further processed through some (non)linear mechanism (for instance Boolean filtering function) and a single (several) bits of the keystream are produced at the time which are then added modulo two to plaintext bits to finally generate the ciphertext. In comparison to block ciphers, the main design goal concerning stream ciphers is either to provide a faster encryption algorithm (faster than block ciphers) or alternatively to have a compact hardware implementation for hardware restricted environments. Two well-known types of shift registers are linear and nonlinear feedback shift registers (shortly LFSR or NFSR). Certain stream cipher designs use LFSRs in combination with a (vectorial) Boolean function whose main goal is to filter the secret state bits and provide a proper confusion of the cipher. Some of the main representative schemes among stream ciphers are SEAL [95, 96], SNOW (see for instance [36]), ISAAC [97], Grain family [47], and many others.

In general, well-designed stream and block cipher only offer computational security, unlike the cryptographic systems which belong to public-key cryptography where commonly the security is related to some well-known hard problem for which no efficient solutions are known. In what follows we briefly describe the main differences between symmetric-key and public-key cryptography. Unlike symmetric key cryptography, where the same secret key is shared between the sender and recipient, the concept of public-key cryptography evolved from an attempt to solve the key distribution by using a public key (known to everyone), and a private (secret) key (known only to the recipient of the message). In a public-key encryption scheme, any person can encrypt a message using the public key of the recipient, but such a message can be decrypted only with the recipient's private key. The security of these systems mainly relies on cryptographic algorithms based on hard mathematical problems that currently admit no efficient solution, such as prime integer factorization, discrete logarithm problem etc.. Moreover, defining the same problem such as discrete logarithm problem on suitable mathematical structures such as elliptic curves may impact positively the hardness of the underlying problem. In addition, public-key encryption algorithms do not require a secure channel for the initial exchange of secret keys between the parties. However, all known public-key cryptosystems are much less efficient than symmetric-key cryptosystems, since they

produce a much lower data throughput (due to the time requirement for encryption). Due to their superior performance in terms of encryption speed in comparison to public key cryptography, symmetric-key encryption schemes are used for encryption of data whereas public key algorithms are mainly employed for key exchange.

With respect to the type of information the adversary has access to, there exist four main classes of cryptanalysis:

- Ciphertext-only attack scenario assumes that the cryptanalyst (attacker) has only passive capability to listen to the encrypted communication. By observing only the ciphertext, the goal of the attacker is to recover the encryption key (or a part of the key), or a portion of the plaintext;
- Known-plaintext attacks regard the scenario when the cryptanalyst tries to recover the key or a part of the key while having some plaintext and the corresponding ciphertext pairs at his disposal;
- Chosen-plaintext attacks presume that the attacker can obtain the ciphertexts for arbitrary plaintexts. The goal of the attack is to recover (portion of) the secret key;
- Chosen-ciphertext scenario assumes that the cryptanalyst has access to the decryption equipment and can decrypt any ciphertext. From these pieces of information the objective is to deduce the key, which can be securely embedded in the equipment, from the ciphertext-plaintext pairs.

From now on we focus our attention on the design and cryptanalysis of stream ciphers and in particular on a subfamily of these schemes that use LFSR/NFSR in combination with filtering (vectorial) Boolean functions.

Among various cryptanalytic techniques applicable to stream ciphers *algebraic attacks* (AA) and *fast algebraic attacks* (FAA) [25, 26] have received a lot of attention. These attacks being generic to LFSR-based stream ciphers have substantially increased the design requests related to the choice of filtering (vectorial) Boolean functions. The core idea behind the two attacks can be summarized as follows. The first step is to set up a low degree algebraic system of multivariate equations in the secret key/state bits, where the degree of these equations is closely related to the algebraic properties of the of the filtering function F (see Figure 8.3). The second step is to solve the system of equations and recover the secret key/state bits. Whereas the second step is well understood, the first step of finding low degree multivariate equations for relatively large number of input variables n is still an open problem due to complexity issues. During the past decade, an efficient evaluation of the resistance of nonlinear Boolean functions against AA and FAA has been addressed in many works due to a great significance of these estimates from both the design and cryptanalysis point of view. At EUROCRYPT 2003, the first algorithm for determining the existence of annihilators of degree d for an arbitrary n-variable Boolean function f (thus finding function g such that $fg = 0$) was proposed in [25]. Its time complexity is about $O(D^3)$ operations, where $D = \sum_{i=0}^{d} \binom{n}{i}$. There have been many attempts to improve the computational efficiency of these estimates [3, 8, 29, 30, 56], but none of the proposed algorithms can handle Boolean functions

with relatively large number of variables, say $n \geq 30$. One important contribution of this thesis is an efficient proposal of probabilistic methods for determining the algebraic properties of Boolean functions for large input spaces n.

Nonlinear filter generator is a typical representative of a hardware oriented design in stream ciphers (see Figure 8.3). It consists of a single linear feedback shift

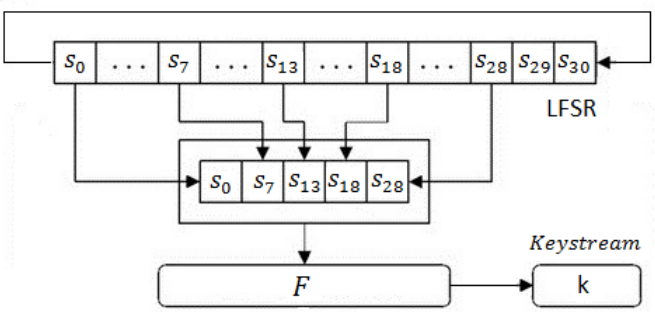

Figure 1.3: Filtering generator with tap positions

register (LFSR) and a nonlinear function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ ($\mathbb{F}_2^n$ is a vector space of all binary vectors of length n) that processes a fixed subset of n stages of the LFSR. This fixed subset of the LFSR's cells is usually called the *taps*. The resistance of nonlinear filter generators against various cryptanalytic attacks, such as (fast) correlation attacks [77, 106, 82], algebraic attacks [23, 24, 78], probabilistic algebraic attacks [9, 87], and attacks that take advantage of the normality of Boolean functions [83], mainly depends on the choice of the filtering function F, and the design rules for ensuring good security margins against these attacks are more or less known today. Nevertheless, *guess and determine cryptanalysis* is a powerful and generic cryptanalytic tool for these schemes which (mainly) does not depend on the filtering function (the same applies to time-memory-data trade-off attacks [7], [48], [54]) but rather to the selection of LFSR: its size, primitive polynomial used and tapping sequence (tap positions used to provide F with the inputs). The main goal of the guess and determine cryptanalysis, when applied to these schemes, is to recover (a part of) the secret state bits contained in the LFSR by exploiting the structure of the cipher. The term "structure" here mainly refers to the tap positions of LFSR used for supplying F with its inputs and the fixed positions of LFSR for implementing a linear recursion through the primitive connection polynomial. It was explicitly stated for the first time in [42] that the choice of tap sequence may play more significant role than the optimization of F in the context of inversion attacks introduced in [42], see also [41, 43]. This important issue of finding (sub)optimal solutions for selecting tap positions, given their number n and the length L of the driving LFSR, appears to be highly neglected in the literature. Although some heuristic approaches have been used for taps selection, an efficient and generic method for this purpose has not been proposed yet. This thesis also contributes in this direction by specifying some algorithms for finding cryptographically (sub)optimal positions of these taps.

Another well known technique in the cryptanalysis of block ciphers is the *differential cryptanalysis* introduced by Eli Biham and Adi Shamir [5]. This technique is

mainly applicable to iterated block ciphers, although it can also be mounted on certain stream ciphers. Basically, differential cryptanalysis is a chosen-plaintext attack though it can be modified into a known-plaintext attack provided that sufficiently many plaintexts are available. In brief, this method searches for plaintext-ciphertext pairs whose difference is constant, and investigates the differential behavior of the cryptosystem, i.e., it exploits the possibility of finding many plaintext pairs with some fixed difference such that the corresponding ciphertext pairs differ by some fixed value. In recent years, differential cryptanalysis has been generalized, resulting in several new techniques such as truncated and higher-order differential cryptanalysis [58, 63], impossible differential cryptanalysis [59], the Boomerang attack [118], and others.

In order to ensure a high security level, functions used in block ciphers need to satisfy various security goals. Among other cryptographic properties (which we briefly describe later on), the concept of linear structures plays an important role in cryptographic applications. Certainly, for functions over finite fields (whose prime field is binary) the substitution boxes (S-boxes) identified as a polynomial $F(x) \in \mathbb{F}_{2^n}[x]$, represented as $F(x) = \sum_{i=0}^{q-1} b_i x^i$, should not contain linear structures a so that $F(x+a) + F(x) = b$ for some fixed $b \in \mathbb{F}_{2^n}$ and for all $x \in \mathbb{F}_{2^n}$. In this case a is called a b-linear structure. Thus, the main problem regarding the linear structures is an identification (or construction) of mappings which do not possess them. A detailed study of the cryptanalytic significance of linear structures was initiated by Evertse [37], where the cryptanalysis of DES-like ciphers was discussed. Linear structures were also considered by Nyberg and Knudsen in the context of provable security against differential attacks [85], and later in many works e.g. [64, 34, 65, 114]. The connection between the existence of linear structures and the differential profile of functions over finite fields is an important area of investigation in the context of the designs of S-boxes. The relevance of this area has increased significantly due to the recent cryptographic need of development of S-boxes (vectorial Boolean functions) suitable for use in lightweight ciphers, see for instance [55, 6]. To sum up the critical technological impact of this area of research we refer to the foreword written by Bart Preneel in the recent book by Tokareva [117] which is entirely devoted to bent functions. Preneel writes: "Perhaps the largest impact on modern cryptography to date would be generated by the study of generalizations to vector Boolean functions that offer strong resistance against differential and linear attacks by Nyberg and others. This work resulted in the S-box used in the Advanced Encryption Standard (AES) that is today used in billions of devices." Incidentally bent functions (on which we elaborate later) are Boolean functions having no linear structures whose cryptographic applications include employment in the designs of CAST, Grain and HAVAL, as well as "non-cryptographic" uses in the designs of Hadamard matrices, strongly regular graphs, Kerdock codes and CDMA sequences.

Apart from linear structures, which have been mentioned in the framework of S-boxes (vectorial Boolean functions), there exist many other indicators which describe the cryptographic properties of a single Boolean function. An *n-variable Boolean function* is a mapping from vector space $\mathbb{F}_2^n$ to binary field with two elements $\mathbb{F}_2 = \{0, 1\}$. One of the fundamental research topics in cryptography is the construction of cryptographically significant Boolean functions, that is a function which possesses

some of the following properties. High *nonlinearity* is one of the most important properties in the design of symmetric-key cryptosystems, since it directly affects the resistance of the cipher to majority of cryptanalytic techniques. The nonlinearity simply measures the Hamming distance to the set of all affine functions. Therefore, a high nonlinearity implies a better resistance to affine approximation attacks [74, 75]. In order to avoid the statistical dependence between the input and output, the concept of *balancedness* implies that a given Boolean function outputs equally many zeros and ones over all possible input values. High *algebraic degree* aims to increase the linear complexity in ciphers. Also, high *algebraic immunity* of order d (that is the minimal degree of annihilator of a given function) plays an important role in providing a high resistance to (Fast) Algebraic attacks on stream ciphers. The resistance of a (block) cipher to differential-like attacks is quantified through derivatives of its S-box, and high resistance to these techniques is achieved with good differential properties.

However, the major problem in construction of cryptographically strong functions is that the multiple criteria mentioned above have to be satisfied at the same time, while there exist intrinsic trade-offs between them. Since the number of Boolean functions in n variables is 2^{2^n}, an exhaustive search of functions which satisfy some of the properties above, is practically impossible (unless the input variable space n is quite small). Thus, bringing new construction methods of these functions is still a vivid research activity.

The term *Bent function* was introduced by Rothaus in 1976 [98], and it is a type of function which has a maximal nonlinearity, i.e., it has a maximal Hamming distance to the set of all affine functions. Since then, this special class of Boolean functions has attracted a lot of attention due to its applications in various areas of mathematics and computer science (for instance in communication systems, sequence design, cryptography, algebraic coding, difference set theory, etc.). There exist various equivalent definitions of bent functions, where the most common uses the Hamming distance as mentioned above, which is actually related to the flat Walsh spectrum (Sylvester-Hadamard transform) of the function (see relation (2.3)). Even though a few generic classes of bent functions have been identified [14, 31, 33, 60] a complete characterization of these functions seems to be elusive. The bent property of vectorial-valued Boolean functions (S-boxes), say $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$, may be extended by requesting that all nonzero linear combinations of the coordinate functions of F are also bent. This means that representing $F(x) = (f_1(x), \ldots, f_m(x))$ as a collection of m Boolean functions f_i, then any nonzero linear combination of the form $a_1 f_1(x) \oplus \ldots \oplus a_m f_m(x)$, where a_i are binary, is again bent. The construction of such vectorial bent functions has been initially considered by Nyberg in [84], where it has been shown that vectorial bent functions can only exist for $m \leq \frac{n}{2}$, and can be constructed using some known classes of bent functions (see for instance MaioranaMcFarland class [31, 32] and the Dillons partial spread class [17, 31, 32, 98]). In the case when $F : \mathbb{Z}_p^n \to \mathbb{Z}_p^n$ and $p > 2$ is a prime number, then instead of the term a vectorial bent we are using the term a planar function.

A generalization of Boolean functions was introduced in [62] and considers a much larger class of mappings from $\mathbb{Z}_q^n$ to $\mathbb{Z}_q$. Nevertheless, due to a more natural connection to cyclic codes over rings, functions from $\mathbb{Z}_2^n$ to $\mathbb{Z}_q$, where $q \geq 2$ is a

positive integer, have drawn even more attention. In [101], K. U. Schmidt studied the relations between generalized bent functions, constant amplitude codes and $\mathbb{Z}_4$-linear codes ($q = 4$). The latter class of mappings is called *generalized bent (gbent) functions* throughout the dissertation. For other generalizations of (bent) Boolean functions we refer reader to [101, 103, 66, 60, 109, 111, 110]. A nice survey on different generalizations of bent functions can be found in [117]. There are several reasons for studying generalized bent (gbent) functions. In the first place, there is a close connection of these objects to standard bent Boolean functions. For instance, the bent conditions imposed on the component functions of gbent functions (using a suitable decomposition) with values in $\mathbb{Z}_q$ has been studied for $q = 4$ [109], $q = 8$ [113], and $q = 16$ [70]. Also, in many other recent works [107, 108, 111, 76] the authors mainly consider the bentness of the component functions for a given prescribed form of gbent functions. A more interesting research challenge in this context is to propose some direct construction methods of functions from $\mathbb{Z}_2^n$ to $\mathbb{Z}_q$ for which a suitable q may give a nontrivial decomposition into standard bent functions, possibly not belonging to the known classes of bent functions. The second reason for the interest in these objects is a close relationship between certain objects used in the design of two different types of communication systems, Orthogonal Frequency-Division Multiplexing (OFDM) [69, 94, 35] and Multi-Carrier Code Division Multiple Access (MC-CDMA) [44, 45, 100]. OFDM is a method of transmitting data simultaneously over equally-spaced carrier frequencies. The method has been proposed for many types of radio systems such as wireless local area networks, digital audio and video broadcasting, Internet networks and 4G mobile communications. MC-CDMA dominates amongst proposals for 3rd Generation cellular communication systems. It is a multiple access scheme used in OFDM-based telecommunication systems, allowing the system to support multiple users at the same time. Both modulation techniques in certain cases suffer from relatively high peak-to-mean envelope power ratio (PMEPR). To overcome these issues, the q-ary sequences lying in complementary pairs [40] (also called Golay sequences) having a low PMEPR can be easily determined from the generalized Boolean function associated with such a sequence, see [104] and the references therein. More precisely, a gbent function corresponds to a q-ary sequence which can reduce the peak-to-average power ratio (PAPR) in such systems to the lowest possible value (called a constant-amplitude code). As a result, some efficient construction methods of gbent functions appear to be very useful in communication systems.

The rest of the thesis is organized as follows. In Chapter 2 the essential background on (generalised) Boolean functions and some basics on guess and determine cryptanalysis is given.

In Chapter 3, a complete characterisation of gbent functions $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$ when q is a power of 2 is deduced, together with some analysis of their dual functions and Gray maps. It turns out that sufficient conditions in this characterisation provide also gbent functions for any even q. Furthermore we discuss a subclass of gbent functions corresponding to relative difference sets which we call $\mathbb{Z}_q$-bent functions, and point out that they correspond to a class of vectorial bent functions. To conclude the chapter, the first general construction methods of gbent functions are proposed.

An optimal selection of tap positions for certain LFSR-based encryption schemes

is investigated from both the design and the cryptanalytic perspective in Chapter 4. Two novel algorithms for an optimal selection of tap positions are given which can be used to provide (sub)optimal resistance to the generic cryptanalytic techniques applicable to these schemes. Two different modes using a variable sampling of keystream blocks are presented, and it is shown that in many cases these modes may outperform the standard GFSGA mode [119] (which is a particular form of guess and determine cryptanalysis). We also demonstrate the possibility of employing GFSGA-like attacks to other design strategies such as NFSR-based ciphers (Grain family for instance [47]) and filter generators outputting a single bit each time the cipher is clocked.

An efficient estimation of the resistance of Boolean functions with relatively large number of inputs against (fast) algebraic attacks is presented in Chapter 5. Based on partial linear relations, a decomposition of nonlinear functions is introduced. This decomposes any given nonlinear Boolean function into linear (affine) subfunctions using disjoint sets of input variables. A general probabilistic decomposition algorithm for nonlinear Boolean functions is presented which gives a new framework for estimating the resistance of Boolean function against (fast) algebraic attacks for large values of n for which the computational complexity of known methods is practically infeasible.

The dissertation is concluded with Chapter 6, where several infinite classes of polynomials which cannot possess linear structures are identified.

The results of this PhD Thesis are published in the following articles:

- S. Hodžić, E. Pasalic. Generalized bent functions - some general construction methods and related necessary and sufficient conditions. *Cryptography and communications*, vol. 7, no. 4, pp. 469-483, 2015.
- S. Hodžić, E. Pasalic. Generalized bent functions - sufficient conditions and related constructions. To appear in *Advances in Mathematics of Communications*. Available at: https://arxiv.org/abs/1601.08084
- S. Hodžić, W. Meidl, E. Pasalic. Full characterization of generalized bent functions as (semi-)bent spaces, their dual and Gray image. *Submitted manuscript*. Available at: https://arxiv.org/abs/1605.05713
- S. Hodžić, E. Pasalic. Construction methods for generalized bent functions. *Submitted manuscript*. Available at: https://arxiv.org/abs/1604.02730
- E. Pasalic, S. Hodžić, S. Bajrić, Y. Wei. Optimizing the placement of tap positions. *International Conference on Cryptography and Information Security in the Balkans, BalkanCryptSec 2014 Turkey, October 16-17*, LNCS 9024, pp. 15–30, 2015.
- S. Hodžić, E. Pasalic, Y. Wei. Optimizing the placement of tap positions and guess and determine cryptanalysis cryptanalysis with variable sampling. *Submitted manuscript*. Available at: https://arxiv.org/abs/1609.08422
- Y. Wei, E. Pasalic, F. Zhang, S. Hodžić. Efficient probabilistic algorithm for estimating the algebraic properties of Boolean functions for large n. *Information Sciences*, vol. 402, pp. 91–104, 2017.

- E. Pasalic, A. Muratović-Ribić, S. Hodžić, S. Gangopathyay. On derivatives of polynomials over finite field through integration. *Discrete Applied Mathematics*, vol. 217, no. 2, pp. 294–303, 2017.

Chapter 2

Boolean functions and filtering generators

In this chapter we cover most of the definitions and concepts related to (generalised) Boolean functions and certain guess and determine attacks. Even though there exist numerous indicators and notions related to (generalised) Boolean functions, we consider only those which will be used in subsequent chapters. In that context, one of the most important tools for analysis of various cryptographic criteria, the so-called Walsh transform, is introduced. Since the formula for the Walsh transform is defined in terms of linear functions, we also recall some known properties of the Sylvester-Hadamard matrix. The chapter is concluded by providing a brief overview of the Filter State Guessing Attack (FSGA) and its generalization (GFSGA), which actually both belong to the class of guess and determine attacks on nonlinear filter generators.

2.1 Boolean functions

Let $\mathbb{F}_q$ denote the Galois field of order $q = p^n$, and let the corresponding vector space be denoted by $\mathbb{F}_p^n$. In the case when $p = 2$, let $\mathbb{F}_2^n$ denote the vector space of binary n-tuples over the finite field with two elements $\mathbb{F}_2 = \{0, 1\}$. We take that the ordering of the space $\mathbb{F}_2^n$ is given as

$$\{(0, 0, \ldots, 0), (1, 0, \ldots, 0), \ldots, (1, 1, \ldots, 1)\},$$

and when the length of the vector is clear from the context we denote the all-zero vector $(0, 0, \ldots, 0)$ by $\mathbf{0}$. By $\mathbb{F}_{2^n}$ we denote the finite Galois field $GF(2^n)$ consisting of 2^n elements. The cyclic group, denoted by $\mathbb{F}_{2^n}^*$, is a multiplicative group consisting of $2^n - 1$ elements which is generated by a primitive element $\alpha \in \mathbb{F}_{2^n}$. Once the basis of the field is fixed, say $\{\gamma_0, \ldots, \gamma_{n-1}\}$ so that $\alpha = \alpha_0\gamma_0 + \ldots + \alpha_{n-1}\gamma_{n-1}$, where $\gamma_i \in \mathbb{F}_{2^n}$ and $\alpha_i \in \mathbb{F}_2$, there is a natural isomorphism between $\mathbb{F}_{2^n}$ and $\mathbb{F}_2^n$ given by

$$\alpha_0\gamma_0 + \ldots + \alpha_{n-1}\gamma_{n-1} \in \mathbb{F}_{2^n} \to (\alpha_0, \ldots, \alpha_{n-1}) \in \mathbb{F}_2^n.$$

We denote the set of integers, real numbers and complex numbers by $\mathbb{Z}$, $\mathbb{R}$ and $\mathbb{C}$, respectively, and the ring of integers modulo r is denoted by $\mathbb{Z}_r$. In some cases, instead of $\mathbb{F}_2^n$ we will write $\mathbb{Z}_2^n$. For $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ in $\mathbb{F}_2^n$, the scalar (or inner) product is defined as $x \cdot y = x_1 y_1 \oplus \ldots \oplus x_n y_n$. The addition over $\mathbb{Z}$, $\mathbb{R}$ and $\mathbb{C}$ is denoted by "+", but also the addition modulo q and it should be understood from the context when reduction modulo q is performed. The binary addition over $\mathbb{F}_2$ is denoted by $\oplus$ in a few cases we use this addition. The cardinality of the set S is denoted by $|S|$. If $z = \mu + \nu i \in \mathbb{C}$, then $|z| = \sqrt{\mu^2 + \nu^2}$ denotes the absolute value of z, and $\overline{z} = \mu - \nu i$ denotes the complex conjugate of z, where $i^2 = -1$, and $\mu, \nu \in \mathbb{R}$. We also denote $\mu = \mathfrak{Re}(z)$ and $\nu = \mathfrak{Im}(z)$.

A *Boolean function* on n variables is any mapping from $\mathbb{F}_2^n$ or $\mathbb{F}_{2^n}$ to $\mathbb{F}_2$, and the set of all such functions is denoted by $\mathcal{B}_n$. Especially, the set of *affine functions* in n variables we define as $\mathcal{A}_n = \{a \cdot x \oplus b \mid a \in \mathbb{Z}_2^n,\ b \in \{0, 1\}\}$. A *vectorial Boolean function* is a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ (or from $\mathbb{F}_{2^n}$ to $\mathbb{F}_{2^m}$).

Representations of Boolean functions

A Boolean function can be represented in several ways, where some of them are addressed in what follows.

The *truth table* of a Boolean function f in n variables is defined as a binary string of values of the function f, i.e.,

$$f = (f(0, 0, \ldots, 0), f(1, 0, \ldots, 0), \ldots, f(1, 1, \ldots, 1)).$$

The Hamming weight of a vector $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$ is defined as $wt(x) = \#\{i : x_i = 1\} = \sum_{i=1}^{n} x_i$, where $\#E$ denotes the cardinality of any set E. Thus, the support of the function f we defined as $supp(f) = \#\{x \in \mathbb{F}_2^n : f(x) = 1\}$. The Hamming distance between two functions $f, g \in \mathcal{B}_n$ is denoted by $d_H(f, g)$ and defined by

$$d_H(f, g) = \#\{x \in \mathbb{F}_2^n : f(x) \neq g(x)\}.$$

Note that the definitions of weights and distance given above are still valid, if $\mathbb{F}_2^n$ is replaced with $\mathbb{F}_{2^n}$. The *nonlinearity* of $f \in \mathcal{B}_n$, denoted by N_f, is defined to be the Hamming distance from the set of all n variable affine functions as

$$N_f = \min_{g \in \mathcal{A}_n} d_H(f, g).$$

Among the classical representations of Boolean functions, a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is commonly represented using its *associated algebraic normal form* (ANF) as

$$f(x_1, \ldots, x_n) = \sum_{u \in \mathbb{F}_2^n} \lambda_u \prod_{i=1}^{n} {x_i}^{u_i}, \tag{2.1}$$

where the variables $x_i \in \mathbb{F}_2$ $(i = 1, \ldots, n)$, $\lambda_u \in \mathbb{F}_2$, $u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$. The *algebraic degree*, denoted by $deg(f)$, is defined as $\max\{wt(u) : \lambda_u \neq 0, u \in \mathbb{F}_2^n\}$. There is a one-to-one correspondence between the truth table and the ANF via so-called inversion formulae.

The *univariate representation* of Boolean functions $f : \mathbb{F}_{2^n} \to \mathbb{F}_2$ is given as

$$f(x) = \sum_{i=0}^{2^n-1} a_i x^i, \;\; a_i \in \mathbb{F}_{2^n}, \tag{2.2}$$

where the coefficients $a_i \in \mathbb{F}_{2^n}$ satisfy the following (Boolean conditions): $a_0, a_{2^n-1} \in \mathbb{F}_2$ and $a_{2i \pmod{2^n-1}} = a_i^2$ for $i = 1, \ldots, 2^n - 2$, due to the condition $f(x)^2 \equiv f(x) \pmod{x^{2^n} - x}$. Consequently, using the univariate representation we formally do not distinguish between $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ and a Boolean mapping $f : \mathbb{F}_{2^n} \to \mathbb{F}_2$. In the case that $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$, then the function F can be viewed as a collection of m Boolean functions, i.e., $F = (f_1(x), \ldots, f_m(x))$, where $f_i : \mathbb{F}_2^n \to \mathbb{F}_2$.

The polynomial degree of $F(x) = \sum_{i=0}^{q-1} b_i x^i$ is the largest i for which $b_i \neq 0$. On the other hand, the algebraic degree of $F(x) = \sum_{i=0}^{q-1} b_i x^i \in \mathbb{F}_q[x]$, where $q = p^n$, is defined as the maximum Hamming weight of the p-adic expansion of the exponent i satisfying that $b_i \neq 0$.

The *derivative* of $f \in \mathcal{B}_n$ at vector $a \in \mathbb{F}_{2^n}$, denoted by $D_a f$, is a Boolean function defined by

$$D_a f(x) = f(x + a) + f(x), \text{ for all } x \in \mathbb{F}_{2^n}.$$

Accordingly, an element $a \in \mathbb{F}_{2^n}^*$ is called a *linear structure* of f if $f(x+a)+f(x) = const. \in \mathbb{F}_2$, for any $x \in \mathbb{F}_{2^n}$.

Walsh Transform

The most significant properties of Boolean functions can be described through the *Walsh-Hadamard transform* (WHT), which for a Boolean function $f \in \mathcal{B}_n$ is defined as

$$W_f(\omega) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \omega \cdot x}, \quad \omega \in \mathbb{F}_2^n. \tag{2.3}$$

The *Walsh spectrum* of the function f is defined as $\{W_f(\omega) : \omega \in \mathbb{F}_2^n\}$. In the case when f is defined on the field $\mathbb{F}_{2^n}$, then in (2.3) the product $\omega \cdot x$ is replaced with $Tr_1^n(\omega x)$ $(\omega, x \in \mathbb{F}_{2^n})$, where Tr_m^n is a *trace function* defined as

$$Tr_m^n(x) = x + x^{2^m} + x^{2^{2m}} + \ldots + x^{2^{(n/m-1)m}}, \quad x \in \mathbb{F}_{2^n}.$$

In other words, it holds that $\omega \cdot x = Tr_1^n(\omega x)$.

The connection between the Walsh transform of a Boolean function $f \in \mathcal{B}_n$ and an arbitrary affine function $g(x) = \omega \cdot x \oplus b$ $(\omega \in \mathbb{F}_2^n, b \in \mathbb{F}_2)$ is given by

$$d_H(f, g) = 2^{n-1} - (-1)^b \frac{W_f(\omega)}{2}.$$

Consequently, the connection between the nonlinearity of the function f and its Walsh transform is given by

$$N_f = 2^{n-1} - \frac{1}{2} \max_{\omega \in \mathbb{F}_2^n} |W_f(\omega)|.$$

Two Boolean functions $f, g \in \mathcal{B}_n$ are said to be a pair of *disjoint spectra functions* [99] if

$$W_f(\omega)W_g(\omega) = 0, \ \textit{for all} \ \omega \in \mathbb{Z}_2^n.$$

(Semi-) Bent functions

The term *bent function* was introduced by Rothaus in 1976 [98], and it is a type of function which has a maximal nonlinearity, i.e., it has a maximal Hamming distance to the set of affine functions. Since this special class of Boolean functions will be used more frequently in later chapters, some of its equivalent properties are summarized as follows.

For a function $f \in \mathcal{B}_n$, the following statements (among other characterizations) are equivalent:

1) The function f is bent;

2) The absolute value of $W_f(\omega)$ is equal to $2^{\frac{n}{2}}$ for all $\omega \in \mathbb{F}_2^n$;

3) The derivative $D_a f(x) = f(x) \oplus f(x \oplus a)$ is balanced for any non-zero $a \in \mathbb{F}_2^n$;

4) The function $f(x) \oplus a \cdot x$ is a bent function for any $a \in \mathbb{F}_2^n$;

5) The matrix $[(-1)^{f(x \oplus y)}]_{x,y \in \mathbb{F}_2^n}$ is a Hadamard matrix;

6) The nonlinearity of f is $N_f = 2^{n-1} - 2^{\frac{n}{2}-1}$.

Recall that a vectorial Boolean function $F(x) = (f_1(x), \ldots, f_m(x))$ is bent if for any nonzero linear combination $a_1 f_1(x) \oplus \ldots \oplus a_m f_m(x)$ $(a_i \in \mathbb{F}_2)$ is a bent Boolean function. Note that the term "vectorial bent function" is used in the binary case, i.e., when $p = 2$. In the case when $n = m$ and $F : \mathbb{F}_{p^n} \to \mathbb{F}_{p^n}$, where $p > 2$ and p is a prime, instead of the term "vectorial bent", the function F is said to be a planar function (mapping).

Besides from having applications in cryptography, one motivation for considering (vectorial) bent functions is their relation to other combinatorial objects. For instance, a vectorial bent function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ $(m \geq 1)$ gives rise to a relative difference set of $\mathbb{F}_2^n \times \mathbb{F}_2^m$. Let G be a group of order nm, let N be a subgroup of G of order m and let R be a subset of G of cardinality r. Then R is called a (n, m, r, λ)-*relative difference set* of G relative to N, if every element $g \in G \setminus N$ can be represented in exactly λ ways as difference $r_1 - r_2$, $r_1, r_2 \in R$, and no nonzero element of N has such a representation.

The *characters of the group* $\mathbb{F}_2^n \times \mathbb{F}_2$ are defined as $\chi_{a,\omega}(x, y) = (-1)^{ay \oplus \omega \cdot x}$, $a \in \{0, 1\}$, $\omega \in \mathbb{F}_2^n$. Note that the Walsh coefficient $W_f(\omega)$ can be also written as $W_f(\omega) = \chi_{1,\omega}(D)$, where $D = \{(x, f(x)) : x \in \mathbb{F}_2^n\}$ is a graph of f.

Relative difference sets can be described with characters as follows (see for instance [115, Section 2.4]).

Let G be an (Abelian) group of order nm and let N be a subgroup of G of order m. A subset R of G (with r elements) is an (n, m, r, λ)- relative difference set of G

relative to N if and only if for every character χ of G

$$|\chi(R)|^2 = \begin{cases} r^2, & \chi = \chi_0 \\ r - \lambda m, & \chi \neq \chi_0, \text{ but } \chi(g) = 1, \ \forall g \in N \\ r, & \text{otherwise.} \end{cases}$$

Since $W_f(u)$ is an integer, for a bent function we have $W_f(u) = 2^{n/2}(-1)^{f^*(u)}$ for a Boolean function $f^* \in \mathcal{B}_n$, called the *dual* of f, which then is also bent. Obviously, Boolean bent functions only exist when n is even.

When n is odd, a *semi-bent* function is defined as a function $f \in \mathcal{B}_n$ for which $\mathcal{W}_f(u) \in \{\pm 2^{\frac{n+1}{2}}, 0\}$ for all $u \in \mathbb{F}_2^n$. A function $f \in \mathcal{B}_n$ is called *s-plateaued* if its Walsh spectrum only takes three values 0 and $\pm 2^{\frac{n+s}{2}}$ $(0 \leq s \leq n)$. Note that n and s must have the same parity.

Many more variants of bent functions, like bent functions in odd characteristic, vectorial bent functions from $\mathbb{F}_p^n$ to $\mathbb{F}_p^m$, negabent functions, bent$_4$ functions, all corresponding to relative difference sets in respective groups, have been investigated. The reader is referred to, for instance, the articles [39, 61, 86, 92, 102, 128] and the recent survey article [93]. For a very general viewpoint considering bent functions over arbitrary Abelian groups, we refer to [91].

2.2 Generalized Boolean functions

We call a function f from $\mathbb{F}_2^n$ to $\mathbb{Z}_q$ $(q \geq 2)$ a *generalised Boolean function*, and denote the set of all such functions by $\mathcal{GB}_n^q$. If $q = 2$, then f is Boolean and $\mathcal{GB}_n^q = \mathcal{B}_n$. I the case of generalized Boolean functions from $\mathbb{F}_2^n$ to $\mathbb{Z}_q$, we prefer to write $\mathbb{Z}_2^n$ instead of $\mathbb{F}_2^n$, since we would like to associate this notation with the corresponding codomain.

To any generalized function $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, for $q = 2^k$, we may associate a unique sequence of Boolean functions $a_i \in \mathcal{B}_n$ $(i = 0, 1, \ldots, k-1)$ such that

$$f(x) = a_0(x) + 2a_1(x) + 2^2 a_2(x) + \ldots + 2^{k-1} a_{k-1}(x), \ \forall x \in \mathbb{F}_2^n. \tag{2.4}$$

In general, the representation (3.25) may be associated to any generalized function with values in $\mathbb{Z}_q$, when $2^{k-1} < q < 2^k$. However, in this case, the representation is not unique.

Having applications of functions from $\mathbb{Z}_2^n$ to $\mathbb{Z}_4$ in code-division multiple access systems in mind, in [101] Schmidt introduced a class of functions which further on were called *generalized bent* (*gbent*). A function $f \in \mathcal{GB}_n^q$ for which the *generalized Walsh-Hadamard transform* (GWHT) at a point $\omega \in \mathbb{Z}_2^n$ defined as the complex valued function

$$\mathcal{H}_f^{(q)}(\omega) = \sum_{x \in \mathbb{Z}_2^n} \zeta_q^{f(x)} (-1)^{\omega \cdot x},$$

where $\zeta_q = e^{2\pi i/q}$ (or any other complex qth-primitive root of unity), has absolute value $2^{n/2}$ for all $\omega \in \mathbb{Z}_2^n$, is called a generalized bent function. Note that when f is Boolean, then $\mathcal{H}_f^{(2)}(u) = W_f(u)$. We recall that in the case of $q = 2^k$ we always have $\mathcal{H}_f^{(2^k)}(u) = 2^{n/2} \zeta_{2^k}^{f^*(u)}$, (except for the case that n is odd and $q = 4$), for a function

$f^* \in \mathcal{GB}_n^{2^k}$, which we call the dual of f, see [70]. As pointed out in [71], f^* is also a gbent function.

Remark 2.2.1 *Throughout the dissertation, at certain places we will also use* normalized Walsh transforms *of Boolean functions and its generalization, that is instead of $W_f(\omega)$ or $\mathcal{H}_f^{(q)}(\omega)$, the Walsh spectrum will contain values $2^{-\frac{n}{2}}W_f(\omega)$ and $2^{-\frac{n}{2}}\mathcal{H}_f^{(q)}(\omega)$. The main reason will be certain connections of these coefficients with rows of the Sylvester-Hadamard matrix, in the case when the underlying functions are bent or gbent, respectively.*

We emphasize here that a gbent function conceptually *does not* correspond to a bent function, since in the definition of GWHT not all characters of $\mathbb{F}_2^n \times \mathbb{Z}_{2^k}$ are considered. Thus, in general, a gbent function does not give rise to a relative difference set. For this reason we extend the definition and introduce the term of a $\mathbb{Z}_q$-bent function. We call a function $f \in \mathcal{GB}_n^{2^k}$ a $\mathbb{Z}_q$-*bent* function if

$$\mathcal{H}_f^{(q)}(\alpha, \omega) = \sum_{x \in \mathbb{Z}_2^n} \zeta_q^{\alpha f(x)} (-1)^{\omega \cdot x}$$

has absolute value $2^{n/2}$ for all $u \in \mathbb{Z}_2^n$ and all nonzero $\alpha \in \mathbb{Z}_{2^k}$.

2.3 Sylvester-Hadamard matrix

In this section we briefly recall the definition of the Sylvester-Hadamard matrix and its certain well-known properties. One additional new property (related to its arbitrary row), which will play an important role in analysis of gbent functions, is provided in Section 3.3.1.

A $(1,-1)$-matrix H of order p is called a *Hadamard* matrix if $HH^T = pI_p$, where H^T is the transpose of H, and I_p is the $p \times p$ identity matrix. A special kind of Hadamard matrix is the *Sylvester-Hadamard* or *Walsh-Hadamard* matrix, denoted by H_{2^k}, which is constructed recursively using Kronecker product $H_{2^k} = H_2 \otimes H_{2^{k-1}}$, where

$$H_1 = (1); \quad H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}; \quad H_{2^k} = \begin{pmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{pmatrix}.$$

For technical reasons we start the row and column index of H_{2^k} with 0, and we denote the r-th row of H_{2^k} by $H_{2^k}^{(r)}$, $0 \le r \le 2^k - 1$. To an integer $j = \sum_{i=0}^{k-1} j_i 2^i$, $0 \le j \le 2^k - 1$, we assign $z_j = (j_0, j_1, \ldots, j_{k-1}) \in \mathbb{F}_2^k$, which also implies an ordering of the elements of $\mathbb{F}_2^k$.

For a function f on $\mathbb{F}_2^n$, the $(1,-1)$-sequence defined by

$$((-1)^{f(v_0)}, (-1)^{f(v_1)}, \ldots, (-1)^{f(v_{2^n-1})})$$

is called the *sequence* of f, where $v_i = (v_{i,0}, \ldots, v_{i,n-1})$, $i = 0, 1, \ldots, 2^n - 1$, denotes the vector in $\mathbb{F}_2^n$ whose integer representation is i, that is, $i = \sum_{j=0}^{n-1} v_{i,j} 2^j$. The vector $v_i = (v_{i,0}, \ldots, v_{i,n-1}) \in \mathbb{F}_2^n$ is uniquely identified by $i \in \{0, 1, \ldots, 2^n - 1\}$.

Several well-known properties of the Sylvester-Hadamard matrices are summarized as follows:

1) Each row of H_{2^k} is uniquely determined by the signs of the entries at positions 2^s, $s = 0, 1, \ldots, k-1$.

2) Let $z_j = (j_0, j_1, \ldots, j_{k-1}) \in \mathbb{F}_2^k$, where $j = \sum_{i=0}^{k-1} j_i 2^i$, $0 \leq j \leq 2^k - 1$. Then

$$H_{2^k}^{(r)} = ((-1)^{z_0 \cdot z_r}, (-1)^{z_1 \cdot z_r}, \ldots, (-1)^{z_{2^k-1} \cdot z_r}),$$

i.e., $H_{2^k}^{(r)}$ is a sequence of a linear function defined on $\mathbb{F}_2^k$.

3) The matrix H_{2^k} is symmetric. Additionally, any two distinct rows are orthogonal, i.e., if $h_{i,j}$ are entries of the i-th row $H_{2^k}^{(i)}$ $(0 \leq i, j \leq 2^k - 1)$, then $\sum_{j=0}^{2^k-1} h_{i,j} h_{p,j} = 0$ if $i \neq j$, and $\sum_{j=0}^{2^k-1} h_{i,j} h_{p,j} = 2^k$ if $i = j$.

2.4 Nonlinear filtering generator

A filtering generator consists of a single LFSR of length L whose n fixed positions (taps) are used as the inputs to a filtering function $F : GF(2)^n \to GF(2)^m$ (also represented as $F(x) = (f_1(x), ..., f_m(x))$), thus outputting $m \geq 1$ keystream bits at the time. A general description of a filter generator is as follows:

$$(z_1^t, \ldots, z_m^t) = (f_1(\ell_n(\mathbf{s}^t)), \ldots, f_m(\ell_n(\mathbf{s}^t))),$$

where $\mathbf{s}^t = (s_0^t, \ldots, s_{L-1}^t)$ is the secret state of the LFSR at time t, the notation $\ell_n(\mathbf{s}^t)$ means that a subset of n bits of $\mathbf{s}^t = (s_0^t, \ldots, s_{L-1}^t)$ (at fixed positions) is passed as the input to Boolean functions $f_1, \ldots, f_m$, and $z_1^t, \ldots, z_m^t$ are the corresponding output keystream bits.

Due to linearity of its feedback connection polynomial, at any $t > 0$ we have $\ell_n(s_0^t, \ldots, s_{L-1}^t) = (\psi_1^t(\mathbf{s}), \ldots, \psi_n^t(\mathbf{s}))$, where the linear functions $\psi_i^t(\mathbf{s}) = \sum_{j=0}^{L-1} a_{i,j}^t s_j$, $(i = 1, \ldots, n)$, are unique linear combinations of the initial secret state bits $\mathbf{s}^0 = (s_0, \ldots, s_{L-1})$, at time $t = 0$. The LFSR is updated by computing the update bit s_L (as a linear combination of $s_0, \ldots, s_{L-1}$ determined by the connection polynomial) and shifting its content to the left (while at the same time outputting the bit s_0), so that $\mathbf{s}^1 = (s_1, \ldots, s_L)$. The binary coefficients $a_{i,j}^t$ above can therefore be efficiently computed from the connection polynomial of LFSR for all $t \geq 0$.

2.4.1 Overview of FSGA and GFSGA

In what follows we briefly describe the main ideas behind FSGA (introduced in [88]) and its extension GFSGA [119]. For both attacks there is no restriction on $F : GF(2)^n \to GF(2)^m$, thus F satisfies all the relevant criteria including a uniform distribution of its preimages.

FSGA description

For every observation of the cipher output $z^t = (z_1^t, \ldots, z_m^t)$ at time t, there are 2^{n-m} possible inputs $x^t \in S_{z^t}$. Moreover, for every guessed preimage $x^t = (x_1^t, \ldots, x_n^t) \in S_{z^t}$, one obtains n linear equations in the secret state bits $s_0, \ldots, s_{L-1}$ through $x_i^t =$

$\sum_{j=0}^{L-1} a_{i,j}^t s_j$, for $1 \leq i \leq n$. The goal of the attacker is to recover the initial state bits $(s_0, \ldots, s_{L-1})$ after obtaining sufficiently many keystream blocks $z^t = (z_1^t, \ldots, z_m^t)$. If the attacker observes the outputs at the time instances $t_1, \ldots, t_c$, so that $nc > L$, then with high probability each system of nc linear equations is independent but only one system will provide a consistent (correct) solution.

As there are $2^{(n-m)c}$ possibilities of choosing c input tuples $(x_1^{t_1}, \ldots, x_n^{t_1}), \ldots, (x_1^{t_c}, \ldots, x_n^{t_c})$, and for each such c-tuple a system of nc linear equations in L variables is obtained. The complexity of solving a single overdefined system of linear equations with L variables is about L^3 operations. Thus, the complexity of the FSGA is about $2^{(n-m)c}L^3$ operations, where $c \approx \lceil \frac{L}{n} \rceil$.

GFSGA description

The major difference to FSGA is that the GFSGA method efficiently utilizes the tap positions of the underlying LFSR. Let the tap positions of the LFSR be specified by the set $\mathcal{I}_0 = \{i_1, i_2, \ldots, i_n\}$, $1 \leq i_1 \leq i_2 \leq \ldots \leq i_n \leq L$. If at the time instance t_1, we assume that the content of the LFSR at these tap positions is given by $(s_{i_1}^{t_1}, \ldots, s_{i_n}^{t_1}) = (a_1, \ldots, a_n)$, then at $t = t_1 + \sigma$ we have $(s_{i_1+\sigma}^{t_1+\sigma}, \ldots, s_{i_n+\sigma}^{t_1+\sigma}) = (a_1, \ldots, a_n)$, where cutting modulo L can be performed if necessary. Notice that the state bits at positions $i_1 + \sigma, \ldots, i_n + \sigma$ does not necessarily intersect with $\mathcal{I}_0$, thus if the intersection is an empty set no information from the previous sampling can be used at the sampling instance $t_1 + \sigma$. However, we can always select σ so that at least one bit of information is conveyed. More formally, the observed outputs at $t_1, \ldots, t_c$, where $t_i = t_1 + (i-1)\sigma$ and $1 \leq \sigma \leq (i_n - i_1)$, may give rise to identical linear equations since the equations $x_i^{t_u} = \sum_{j=0}^{L-1} a_{i,j}^{t_u} s_j$ (where $1 \leq i \leq n$) may be shifted to $x_l^{t_v} = \sum_{j=0}^{L-1} a_{i,j}^{t_v} s_j$, for some $1 \leq i < l \leq n, 1 \leq u < v \leq c$.

It is of importance to determine how many identical linear equations will be obtained for all the sampling instances $t_1, \ldots, t_c$. By introducing $k = \lfloor \frac{i_n - i_1}{\sigma} \rfloor$, and for $\mathcal{I}_0 = \{i_1, i_2, \ldots, i_n\}$ defining recursively:

$$\begin{aligned} \mathcal{I}_1 &= \mathcal{I}_0 \cap \{i_1 + \sigma, i_2 + \sigma, \ldots, i_n + \sigma\}, \\ \mathcal{I}_2 &= \mathcal{I}_1 \cup \{\mathcal{I}_0 \cap \{i_1 + 2\sigma, i_2 + 2\sigma, \ldots, i_n + 2\sigma\}\}, \\ &\vdots \\ \mathcal{I}_k &= \mathcal{I}_{k-1} \cup \{\mathcal{I}_0 \cap \{i_1 + k\sigma, i_2 + k\sigma, \ldots, i_n + k\sigma\}\}. \end{aligned} \tag{2.5}$$

the analysis in [119] showed that the complexity of the GFSGA is closely related to the parameter $r_i = \#\mathcal{I}_i$, where $i = 1, \ldots, k$.

Remark 2.4.1 *For instance, the above notation means that for some $i \in \mathcal{I}_1$ (and therefore $i \in \mathcal{I}_0$) the state bit $s_i^{t_2}$ was used in the previous sampling since it was at the position $i - \sigma \in \mathcal{I}_0$ at time t_1, where $t_2 = t_1 + \sigma$. The idea is easily generalized for $\#\mathcal{I}_i = r_i$, where $i = 2, \ldots, k$.*

The number of identical equations obtained in [119] is given as follows. If $c \leq k$, then in total $\sum_{i=1}^{c-1} r_i$ identical linear equations are obtained, whereas for $c > k$ this number is $\sum_{i=1}^{k} r_i + (c - k - 1)r_k$. Note that in this case $r_k = r_{k+1} = \cdots = r_{c-1}$ due

to the definition of k, which simply guarantees that after k sampling instances the maximum (and constant) number of repeated equations is attained. Consequently, the time complexity of the attack for $c \leq k$ was estimated as,

$$\begin{aligned} T_{Comp.}^{c \leq k} &= 2^{(n-m)} \times 2^{(n-m-r_1)} \times \ldots \times 2^{(n-m-r_{(c-1)})} \times L^3 \\ &= 2^{(n-m)c - \sum_{i=1}^{c-1} r_i} \times L^3, \end{aligned} \tag{2.6}$$

and similarly, if $c > k$, the time complexity for $c > k$ was given by

$$\begin{aligned} T_{Comp.}^{c > k} &= 2^{(n-m)} \times 2^{(n-m-r_1)} \times \ldots \times \\ &\times \; 2^{(n-m-r_k)} \times 2^{(n-m-r_k) \times (c-k-1)} \times L^3 \\ &= 2^{(n-m)c - (\sum_{i=1}^{k} r_i + (c-k-1) r_k)} \times L^3. \end{aligned} \tag{2.7}$$

Remark 2.4.2 *If $n - m - r_i \leq 0$, for some $i \in \{1, \ldots, k\}$, then the knowledge of these r_i bits allows the attacker to uniquely identify the exact preimage value form the set of 2^{n-m} possible preimages, i.e., we assume $2^{(n-m-r_i)} = 1$ when $n - m - r_i \leq 0$.*

Chapter 3

Generalised bent (gbent) functions

In this chapter, we address the important problem of specifying the conditions that $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$ is a generalized bent (gbent) function. Results of this chapter are published in [50, 52, 51, 53].

When $q = 4$ and n is even, from [109] we have that a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_4$, given in the form $f(x) = a_0(x) + 2a_1(x)$, is gbent if and only if a_1 and $a_1 \oplus a_0$ are bent Boolean functions. Several other results related to the case $q = 4$ and n even are given in [101], where some of them involve the trace forms of Galois rings whose employment is also discussed in [123]. For the octal case $q = 8$, both necessary and sufficient conditions for the component functions of $f : \mathbb{Z}_2^n \to \mathbb{Z}_8$, representing uniquely f as $f(x) = a_0(x)+2a_1(x)+2^2a_2(x)$ where a_0, a_1, a_2 are Boolean functions, were given in [111]. Some recent results on gbent functions related to the case $q = 8$ can be found in [113, 76]. Similar conditions for $q = 16$ are obtained in [70]. In addition, the Walsh spectra of these functions must satisfy certain conditions related to Hadamard matrices which makes the design methods rather involved. In difference to the previous work [111, 113, 70], where the sufficient and necessary conditions when $q = 4, 8, 16$ were derived, we consider the general case of q being a power of 2 and subsequently derive necessary and sufficient conditions for f to be gbent. Additionally, our sufficient conditions provide gbent functions for any even $q \geq 2$. These conditions are equivalent to those very recently published online in [116]. Notably we then describe gbent functions as algebraic objects, a characterization which goes far beyond the conventional descriptions in terms the Walsh transforms of linear combinations of the coordinate functions, which in accordance with the terminology for vectorial bent function we call the component functions of the gbent function. We show that gbent functions correspond to affine spaces of bent functions when n is even and semi-bent functions when n is odd, with certain interesting additional properties, which we precisely describe. Employing conventional equivalence, we show that gbent functions and affine spaces of bent (semi-bent) functions with these properties are identical objects. These results essentially completely resolve the case of gbent functions from $\mathbb{Z}_2^n$ to $\mathbb{Z}_{2^k}$, using the approach based on Hadamard matrices

introduced in [52]. We emphasize the fact that the sufficient and necessary conditions for $q \in \{4, 8\}$ were derived in a nontrivial manner employing so-called Jacobi sums and the same technique could not be applied for larger q of the form 2^k.

The whole approach and the sufficient conditions derived here is based on an alternative characterization and computation of the generalized Walsh-Hadamard spectral values through using the standard Walsh spectra of the component Boolean functions a_i when $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$ is (uniquely) represented by relation (3.25), i.e., as $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-1}a_{k-1}(x)$ for $q = 2^k$. We note that q being a power of 2 is the most interesting case due to applications. However, it turns out that our approach is not so efficient when considering even q in the range $2^{k-1} < q < 2^k$. To give some sufficient conditions for the gbent property in this case we were forced to consider a different form of f which necessarily contains the coefficient $q/2$ in its representation. Thus, in this case (again to avoid some difficult character sums) the function f is rather represented as $f(x) = \frac{q}{2}a(x) + a_0(x) + 2a_1(x) + \cdots + 2^{k-2}a_{k-2}(x)$ which then simplifies the analysis of their properties. Using these representations we derive a compact and simple formula to compute the generalized Walsh-Hadamard spectra in terms of the spectra of the component functions of f. Based on this formula some sufficient conditions for the gbent property are derived which in turn gives us the possibility to specify certain generic classes of gbent functions.

Several more general classes of gbent functions were described in [111], such as generalized Maiorana-McFarland class (GMMF) [111, Theorem 8], generalized Dillon class (GD)[111, Theorem 9], partial spread class (PS) [71] and generalized spread class (GS) [111, Theorem 10]. It has been shown that the GD and GMMF classes are both contained in the GS class [111, Theorem 12]. The construction of these gbent functions was also considered in [107], though from the cross-correlation point of view. However, the main limitations related to the previously mentioned general classes (with the exception of GMMF) is that they only provide sufficient gbent conditions which are not easy to satisfy in an efficient manner. Although the gbent functions from the GMMF class are easily constructed, they are defined only on even number of variables.

Based on the necessary and sufficient conditions which we derive, in Section 3.4 we present the fist generic method for construction of gbent functions for any even q when n is even and for $q = 2^r$ when n is odd. The method is based on the use of the Maiorana-McFarland (MM) class of functions which contains both semi-bent and bent functions. Nevertheless, the difficulty lies in the fact that the component functions (more precisely certain linear combinations of them) apart from being bent or semi-bent (depending on the parity of n) must satisfy additional constraints. More precisely, when n is odd certain linear combinations of the component functions must be disjoint spectra semi-bent functions and apart from that the signs of their Walsh coefficients are supposed to satisfy certain Hadamard recursion. Therefore, the selection of component functions turns out to be a rather nontrivial task. We efficiently solve this problem by using suitable permutations for deriving disjoint spectra semi-bent functions from the MM class that satisfy the gbent conditions. The question of finding another generic methods for the same purpose is left as an interesting open problem. We emphasize that the case n even which is also briefly discussed is of minor importance (due to the generic method provided through the

GMMF class) and the main contribution is a novel and efficient method of satisfying rather demanding gbent conditions when n is odd. At the end, we analyze the class of gbent functions of the form $g(x) = \frac{q}{2}a(x) + kb(x)$, $k \in \{\frac{q}{4}, \frac{3q}{4}\}$, $q = 4s$ ($s \in \mathbb{N}$), where we show that certain constructions of gbent functions for $q \in \{4, 8\}$ [107, 113, 111] belong to this class of functions. We note that many gbent functions constructed by the previously mentioned generic method (which uses MM class of Boolean functions) do not have the form $\frac{q}{2}a(x) + kb(x)$, since it clearly has many equal or zero coordinate functions (in comparison to the full form given by (3.25)).

The rest of this chapter is organized as follows. In Section 3.1, a new convenient formula for computing the generalized Walsh-Hadamard spectra of $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$ in terms of the spectral values of its component functions, is derived. Sufficient conditions, given in terms of Hadamard matrices, for a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$ with any even q, to be gbent are given in Section 3.2. This section is further extended in terms of necessary conditions by Section 3.3.2, where we provide a complete characterization of gbent functions when q is a power of 2. Additionally, the notion of $\mathbb{Z}_q$-bent functions is introduced and analyzed, as well as the dual (n even case) and the Gray image of a gbent function. The first generic construction methods of gbent functions (for any n, and even q) are given in Section 3.4, where in Section 3.4.6 we illustrate certain construction details for n odd case. The special class of gbent functions of the form $\frac{q}{2}a(x) + kb(x)$, $k \in \{\frac{q}{4}, \frac{3q}{4}\}$ is analyzed in Section 3.5, where we show that most of the know construction for $q \in \{4, 8\}$ belong to this class.

3.1 Motivation and Conjecture on GWHT

In this section, we recall some results related to quaternary and octal gbent functions [109, 113] in terms of GWHT. The necessary and sufficient conditions for gbent property derived in [109, 113] for $q = 4$ and $q = 8$ motivates us to conjecture that similar sufficient conditions are valid for arbitrary even q, which is then proved in Section 3.2.3. Notice that proving the necessity of these conditions turns out to be hard, although there are certain indications that the sufficient conditions given in Theorem 3.2.1 are also necessary.

Remark 3.1.1 *In this section and Section 3.2 we use the normalized Walsh transforms for Boolean and its generalization (Remark 2.2.1), since it will emphasize the close connection between the conditions for gbent property and Sylvester-Hadamard matrices.*

If $2^{k-1} < q \leq 2^k$, to any generalized function $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, we may associate a (unique) sequence of Boolean functions $a_i \in \mathcal{B}_n$ $(i = 0, 1, \ldots, k-1)$ such that

$$f(x) = a_0(x) + 2a_1(x) + 2^2 a_2(x) + \ldots + 2^{k-1}a_{k-1}(x), \ \forall x \in \mathbb{Z}_2^n. \tag{3.1}$$

The functions $a_i(x)$, $i = 0, 1, \ldots, k-1$, are called the component functions of the function $f(x)$. When $q = 4$ it was shown that the function $f(x) = a_0(x) + 2a_1(x)$, $a_0, a_1 \in \mathcal{B}_n$, is gbent if and only if $a_1(x)$ and $a_0(x) \oplus a_1(x)$ are bent Boolean functions [109]. Note that the last condition implies that $a_0(x)$ is not necessarily bent (it can be affine for instance), and consequently only $a_1(x)$ needs to be bent. In addition, the

GWHT of the function f in this case is expressed in terms of the WHT transforms of the functions $a_1(x)$ and $a_0(x) \oplus a_1(x)$, i.e., we have

$$\mathcal{H}_f(u) = \frac{1}{2}[(W_{a_1}(u) + W_{a_0 \oplus a_1}(u)) + i(W_{a_1}(u) - W_{a_0 \oplus a_1}(u))], \ \forall u \in \mathbb{Z}_2^n.$$

However, we may rewrite this equality so that we view $\mathcal{H}_f$ as a linear combination of W_{a_1} and $W_{a_0 \oplus a_1}$, where the coefficients are complex numbers, that is,

$$\mathcal{H}_f(u) = \frac{1}{2}(1+i)\, W_{a_1}(u) + \frac{1}{2}(1-i)\, W_{a_0 \oplus a_1}(u). \tag{3.2}$$

In the case when $q = 8$, for $f \in \mathcal{GB}_n^8$ given by

$$f(x) = a_0(x) + 2a_1(x) + 2^2 a_2(x), \tag{3.3}$$

the GWHT of f is given by the following lemma.

Lemma 3.1.2 *[111, 113] Let $f \in \mathcal{GB}_n^8$ as in (3.3). Then,*

$$4\mathcal{H}_f(u) = \alpha_0 W_{a_2}(u) + \alpha_1 W_{a_0 \oplus a_2}(u) + \alpha_2 W_{a_1 \oplus a_2}(u) + \alpha_3 W_{a_0 \oplus a_1 \oplus a_2}(u), \tag{3.4}$$

where $\alpha_0 = 1 + (1+\sqrt{2})i$, $\alpha_1 = 1 + (1-\sqrt{2})i$, $\alpha_2 = 1 + \sqrt{2} - i$, $\alpha_3 = 1 - \sqrt{2} - i$.

Remark 3.1.3 *A special case of selecting $a_0(x) = 0$ appears to be interesting. In the first place, the condition relating the Walsh coefficients becomes simpler, that is,*

$$4\mathcal{H}_f(u) = 2(1+i)W_{a_2}(u) + 2(1-i)W_{a_1 \oplus a_2}(u), \ \forall u \in \mathbb{Z}_2^n.$$

Then, assuming further that $a_1(x) = 0$ would actually give $4\mathcal{H}_f(u) = 4W_{a_2}(u)$, meaning that we only have one bent function and that the function $f(x) = 4a_2(x)$ is gbent though its codomain only takes the values from the set $\{0, 4\}$. In general, any function defined as $f(x) = \frac{q}{2}a(x)$ is gbent if and only if $a(x)$ is a bent function.

Remark 3.1.4 *Apart form the trivial case discussed in Remark 3.1.3, we may also consider other suitable choices for the component functions a_0, a_1 and a_2. Fixing a_2 to be bent we may consider $a_0, a_1 \in \mathcal{A}_n$ to be suitably chosen affine functions so that the above conditions are satisfied. Indeed, since a_2 being bent implies that the addition of any affine function to it does not affect the bent property we can assume that $a_i \in \mathcal{A}_n$ for $i = 0, 1$. It is well-known that for $a_i(x) = a_{i,0} + a_{i,1}x_1 + \ldots + a_{i,n}x_n$, if the Walsh transform of $f(x)$ at point u is $W_f(u)$ then the transform of $f(x) + a_i(x)$ at point u is $(-1)^{a_{i,0}} W_f(u + a^{(i)})$, where $a^{(i)} \in \mathbb{Z}_2^n$ is given as $a^{(i)} = (a_{i,1}, \ldots, a_{i,n})$. Hence, (3.4) can be rewritten as,*

$$4\mathcal{H}_f(u) = \alpha_0 W_{a_2}(u) + \alpha_1 (-1)^{a_{0,0}} W_{a_2}(u + a^{(0)}) + \alpha_2 W_{a_1 \oplus a_2}(u) + \alpha_3 (-1)^{a_{0,0}} W_{a_1 \oplus a_2}(u + a^{(0)}).$$

Notice that $\mathcal{H}_f$ in (3.4) is again a linear combination of the WHTs of the functions $a_2(x)$, $a_0(x) \oplus a_2(x)$, $a_1(x) \oplus a_2(x)$, $a_0(x) \oplus a_1(x) \oplus a_2(x)$. Moreover, the following theorem imposes the conditions for the function $f \in \mathcal{GB}_n^8$ to be a gbent function.

Theorem 3.1.5 *[111] Let $f \in \mathcal{GB}_n^8$ as in (3.3). Then:*

1) If n is even, then f is generalized bent if and only if a_2, $a_0 \oplus a_2$, $a_1 \oplus a_2$, $a_0 \oplus a_1 \oplus a_2$ are all bent, and

$$(*)\ W_{a_0 \oplus a_2}(u) W_{a_1 \oplus a_2}(u) = W_{a_2}(u) W_{a_0 \oplus a_1 \oplus a_2}(u), \text{ for all } u \in \mathbb{Z}_2^n;$$

2) If n is odd, then f is generalized bent if and only if a_2, $a_0 \oplus a_2$, $a_1 \oplus a_2$, $a_0 \oplus a_1 \oplus a_2$ are semi-bent satisfying

$$(**) : W_{a_0 \oplus a_2}(u) = W_{a_2}(u) = 0 \quad \wedge \quad |W_{a_1 \oplus a_2}(u)| = |W_{a_0 \oplus a_1 \oplus a_2}(u)| = \sqrt{2}; \text{ or}$$
$$W_{a_1 \oplus a_2}(u) = W_{a_0 \oplus a_1 \oplus a_2}(u) = 0 \quad \wedge \quad |W_{a_0 \oplus a_2}(u)| = |W_{a_2}(u)| = \sqrt{2},$$

for all $u \in \mathbb{Z}_2^n$.

In general, a formula which gives the GWHT of the function f given by (3.25) is given by the following theorem.

Theorem 3.1.6 *[111, 113] The Walsh-Hadamard transform of $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, $2^{k-1} < q \leq 2^k$, where $f(x) = \sum_{i=0}^{k-1} a_i(x) 2^i$, $a_i \in \mathcal{B}_n$ is given by*

$$\mathcal{H}_f(u) = 2^{-k} \sum_{I \subseteq \{0,\ldots,k-1\}} \zeta^{\sum_{i \in I} 2^i} \sum_{J \subseteq I, K \subseteq \overline{I}} (-1)^{|J|} W_{\sum_{t \in J \cup K} a_t(x)}(u). \tag{3.5}$$

This implicit expression does not reveal the fact that $\mathcal{H}_f$ of a function f represented as in (3.25) can be given explicitly as a linear combination (with complex coefficients that can be efficiently computed) of the WHTs of some linear combinations of its component functions $a_i(x)$, $i = 0, 1, \ldots, k-1$. Therefore, for an arbitrary generalized Boolean function f given by (3.25), it is of great importance to develop a more useful formula for its GWHT which will be given in the next section.

Before we state our conjecture regarding the GWHT and the conditions $(*)$-$(**)$ in general, we first formalize our observations. Let $\Theta_i(x)$ be the function defined as

$$\Theta_i(x) = (-1)^{z_{i,0} a_0(x) \oplus z_{i,1} a_1(x) \oplus \ldots \oplus z_{i,k-1} a_{k-1}(x)}, \tag{3.6}$$

where $z_i = (z_{i,0}, z_{i,1}, \ldots, z_{i,k-1}) \in \mathbb{Z}_2^k$ and i denotes its integer representation, $i = 0, \ldots, 2^k - 1$.

Remark 3.1.7 *Note that the function $\Theta_i(x)$ actually gives (-1) powered to all possible linear combinations of the component functions $a_0(x), a_1(x), \ldots, a_{k-1}(x)$. In addition, we always have $\zeta^{\frac{q}{2} a_{k-1}(x)} = (-1)^{a_{k-1}(x)}$ for $q = 2^k$.*

For $q = 8 = 2^3$, thus $k = 3$, let us consider $f : \mathbb{Z}_2^n \to \mathbb{Z}_8$ given by (3.3). Since $\zeta^{4a_2(x)} = (-1)^{a_2(x)}$, the GWHT is given as:

$$\mathcal{H}_f(u) = \sum_{x \in \mathbb{Z}_2^n} \zeta^{f(x)} (-1)^{u \cdot x} = \sum_{x \in \mathbb{Z}_2^n} \zeta^{a_0(x) + 2a_1(x)} (-1)^{a_2(x) \oplus u \cdot x}. \tag{3.7}$$

Hence, for $q = 8$ we have $z = (z_0, z_1) \in \mathbb{Z}_2^2$, $\Theta_z(x) = (-1)^{z_0 a_0(x) \oplus z_1 a_1(x)}$, where

$$\begin{aligned} \Theta_0(x) = \Theta_{(0,0)}(x) &= 1, \\ \Theta_1(x) = \Theta_{(1,0)}(x) &= (-1)^{a_0(x)}, \\ \Theta_2(x) = \Theta_{(0,1)}(x) &= (-1)^{a_1(x)}, \\ \Theta_3(x) = \Theta_{(1,1)}(x) &= (-1)^{a_0 \oplus a_1(x)}, \end{aligned} \tag{3.8}$$

and

$$\zeta^{a_0(x)+2a_1(x)} = 2^{-2}(\alpha_0\Theta_0(x) + \alpha_1\Theta_1(x) + \alpha_2\Theta_2(x) + \alpha_3\Theta_3(x)),$$

where α_i are given in Lemma 3.1.2.

From the above one can find that for $q = 8$ (similarly when $q = 4$) we have that $\zeta^{f(x)}$ can be represented as a complex linear combination of the functions $\Theta_i(x)$ with the possibility of computing the complex coefficients α_i efficiently. Thus, it may be conjectured that this representation is valid in general for arbitrary q which is shown in Theorem 3.1.9 in the next subsection. This result is proved useful later for deriving sufficient conditions of gbent property and for generalizing Theorem 3.1.5 though covering all values of q, when q is even.

3.1.1 New GWHT formula

In this section, we derive a new GWHT formula for any generalized function $f \in \mathcal{GB}_n^q$ which computes $\mathcal{H}_f$ by using the Walsh spectral values of the component functions and the coefficients α_i.

Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, $2^{k-1} < q \leq 2^k$, where again $f(x) = a_0(x) + 2a_1(x) + \ldots + 2^{k-1}a_{k-1}(x)$, $a_i(x) \in \mathcal{B}_n$. For convenience, we introduce the coefficients $c_i = 2^i$, for $i = 0, \ldots, k-1$, thus writing $f(x) = \sum_{i=0}^{k-1} c_i a_i(x)$. Notice that whatever formal representation of f is used (see also Example 3.1.8), once the function f has been specified in terms of its input and output values, the decomposition into the Boolean component function $a_i(x)$ as given above is unique and any other representation can be transformed into this form.

Assume now that ζ^f can be represented as a linear combination of the functions $\Theta_i(x)$ as

$$\zeta^{f(x)} = \zeta^{\sum_{i=0}^{k-1} c_i a_i(x)} = \sum_{i=0}^{2^k-1} \alpha_i \Theta_i(x), \tag{3.9}$$

for some complex numbers $\alpha_i \in \mathbb{C}$ and $\Theta_i(x) = (-1)^{z_{i,0}a_0(x) \oplus \cdots \oplus z_{i,k-1}a_{k-1}(x)}$, as given by (3.6). The main task is to find the coefficients α_i such that (3.9) holds for every $x \in \mathbb{Z}_2^n$.

Consider an arbitrary but fixed $x' \in \mathbb{Z}_2^n$ such that $(a_0(x'), \ldots, a_{k-1}(x')) = z_j \in \mathbb{Z}_2^k$, where j is the integer representation of a binary vector z_j. To relate the functions Θ_i to the rows (columns) of the Hadamard matrix we need the following useful identification. It is well-known that the rows of the Hadamard matrix H_{2^k} of size $2^k \times 2^k$ are the evaluations of all linear functions in $\mathcal{B}_k$, that is, the j-th row of H_{2^k} (alternatively the j-th column since $H_{2^k} = H_{2^k}^T$) can be expressed as $H_{2^k}^{(j)} = \{(-1)^{z_j \cdot y} \mid y \in \mathbb{Z}_2^k\}$, where z_j is fixed. Therefore,

$$(\Theta_0(x'), \Theta_1(x'), \ldots, \Theta_{2^k-1}(x')) = H_{2^k}^{(j)}.$$

Indeed, for a fixed $x' \in \mathbb{Z}_2^n$ the value of a binary vector $(a_0(x'), \ldots, a_{k-1}(x')) = z_j$ is also fixed and it is easy to verify that,

$$(\Theta_0(x'), \Theta_1(x'), \ldots, \Theta_{2^k-1}(x')) = ((-1)^{z_j \cdot z_0}, (-1)^{z_j \cdot z_1}, \ldots, (-1)^{z_j \cdot z_{2^k-1}}) = H_{2^k}^{(j)},$$

where $z_0, z_1, \ldots, z_{2^k-1}$ are elements of the set $\mathbb{Z}_2^n$. Furthermore, for this particular (but arbitrary) value x' the fact that $(a_0(x'), \ldots, a_{k-1}(x')) = z_j$ implies that

$$\zeta^{f(x')} = \zeta^{\sum_{i=0}^{k-1} c_i a_i(x')} = \zeta^{z_j \odot (c_0, \ldots, c_{k-1})}. \tag{3.10}$$

Now, if we define the column matrix $\Lambda = [\alpha_i]_{i=0}^{2^k-1}$ to be a matrix of the coefficients α_i, the previous discussion together with (3.9) implies that

$$H_{2^k}^{(j)} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{2^k-1} \end{pmatrix}_{2^k \times 1} = H_{2^k}^{(j)} \Lambda = \zeta^{z_j \odot (c_0, \ldots, c_{k-1})}.$$

Notice that when z_j goes through $\mathbb{Z}_2^k$ the value $z_j \odot (c_0, \ldots, c_{k-1})$ goes through $\mathbb{Z}_q$, since the operation $\odot$ means cutting by modulo q. Therefore, it is convenient to define a column matrix B as a matrix of all corresponding powers of ζ, that is, $B = [\zeta^{z_i \odot (c_0, \ldots, c_{k-1})}]_{i=0}^{2^k-1}$ or given in the matrix form as,

$$B = \begin{pmatrix} \zeta^0 \\ \zeta^{c_0} \\ \vdots \\ \zeta^{c_0 + \cdots + c_{2^k-1}} \end{pmatrix}. \tag{3.11}$$

And obviously assuming (3.9) is valid the following system of equations must be satisfied

$$H_{2^k} \Lambda = B. \tag{3.12}$$

As mentioned previously, the function $f \in \mathcal{GB}_n^q$ may be given in different forms, for instance $f(x) = \sum_{i=0}^{d} c_i b_i(x)$, where $b_i \in \mathcal{B}_n$ but $c_i \in \mathbb{Z}_q$ and in general $c_i \neq 2^i$. Nevertheless, one can easily transform such a function into the form discussed above. Note that the solution Λ of the system (3.12) implies that the equality (3.9) holds for any $x \in \mathbb{Z}_2^n$. The main reason for this is the fact that the Hadamard matrix covers all possible values of the vector $(\Theta_0(x), \Theta_1(x), \ldots, \Theta_{2^k-1}(x))$. Therefore, for any $x \in \mathbb{Z}_2^n$ the evaluation of the component functions $(a_0(x), \ldots, a_{k-1}(x))$ implies that the corresponding Hadamard row multiplied with Λ will always be equal to the corresponding power of ζ.

Since the determinant of the Sylvester-Hadamard matrix is given as $\det(H_{2^k}) = \pm 2^{k2^{k-1}}$, using the fact that $H_{2^k}^{-1} = 2^{-k} H_{2^k}^T$ (H_{2^k} is symmetric), we have that the unknown column matrix $\Lambda = [\alpha_i]_{i=0}^{2^k-1}$ is (uniquely) given by

$$\Lambda = H_{2^k}^{-1} B = 2^{-k} H_{2^k}^T B = 2^{-k} H_{2^k} B. \tag{3.13}$$

In the following example, we illustrate a complete procedure of finding α_i with respect to both discussed representations of the function $f(x)$.

Example 3.1.8 *Let us consider generalized function $f(x) = 2a_0(x) + 3a_1(x)$, for $q = 6$. Since we have only two component functions $a_0, a_1 \in \mathcal{B}_n$, it means that we may consider the system of equations given (in the matrix form) as $H_{2^2}\Lambda = B$, where the matrix B (defined by (3.11)) is given as*

$$B = \begin{pmatrix} 1 \\ \zeta^2 \\ \zeta^3 \\ \zeta^5 \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{2} + \frac{i\sqrt{3}}{2} \\ -1 \\ \frac{1}{2} - \frac{i\sqrt{3}}{2} \end{pmatrix}.$$

Consequently, the matrix of coefficients $\Lambda = [\alpha_i]_{i=0}^3$ is given by $\Lambda = 2^{-2}H_{2^2}B$, i.e.,

$$\Lambda = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = 2^{-2}\begin{pmatrix} 0 \\ 0 \\ 1 + i\sqrt{3} \\ 3 - i\sqrt{3} \end{pmatrix}.$$

In addition, for all vectors $z_i = (z_{i,0}, z_{i,1}) \in \mathbb{Z}_2^2$, functions $\Theta_i(x) = (-1)^{z_{i,0}a_0(x) \oplus z_{i,1}a_1(x)}$ are given by

$$\Theta_0(x) = 1, \;\; \Theta_1(x) = (-1)^{a_0(x)}, \;\; \Theta_2(x) = (-1)^{a_1(x)}, \;\; \Theta_3(x) = (-1)^{a_0(x) \oplus a_1(x)}.$$

Hence, the term $\zeta^{f(x)}$ can be decomposed as:

$$\begin{aligned} \zeta^{f(x)} &= \sum_{i=0}^{2^2-1} \alpha_i \Theta_i(x) = 2^{-2}(0 \cdot \Theta_0(x) + 0 \cdot \Theta_1(x) + (1 + i\sqrt{3})\Theta_2(x) + (3 - i\sqrt{3})\Theta_3(x)) \\ &= 2^{-2}((1 + i\sqrt{3})(-1)^{a_1(x)} + (3 - i\sqrt{3})(-1)^{a_0(x) \oplus a_1(x)}) = \begin{cases} 1, & (a_0(x), a_1(x)) = (0,0) \\ \zeta^2, & (a_0(x), a_1(x)) = (1,0) \\ \zeta^3, & (a_0(x), a_1(x)) = (0,1) \\ \zeta^5 & (a_0(x), a_1(x)) = (1,1) \end{cases}. \end{aligned}$$

In the above computation $f(x) = 2a_0(x) + 3a_1(x)$ was not written in the form (3.25). We can rewrite f in the form (3.25) as $f(x) = b_0(x) + 2b_1(x) + 4b_2(x)$ for some component functions $b_0, b_1, b_2 \in \mathcal{B}_n$, since we have $2^2 < q \leq 2^3$ ($q = 6$). In that case, we would consider the system $H_{2^3}\Lambda' = B'$, where $\Lambda' = [\alpha_i']_{i=0}^{2^3-1}$ and $B' = [\zeta^i]_{i=0}^{2^3-1}$ (B' contains all powers of ζ). One may notice that the only difference in considering the function f as $2a_0 + 3a_1$ and $b_0 + 2b_1 + 4b_2$ is in the size of corresponding systems and definition of the matrices B and B'.

Hence, from (3.13) we have $\alpha_i = 2^{-k}H_{2^k}^{(i)}B$, for $i = 0, \ldots, 2^k - 1$, and together with (3.9) we have that the GWHT is given as

$$\mathcal{H}_f(u) = \sum_{x \in \mathbb{Z}_2^n} \zeta^{f(x)}(-1)^{u \cdot x} = \sum_{x \in \mathbb{Z}_2^n} \left((-1)^{u \cdot x} \sum_{i=0}^{2^k-1} \alpha_i \Theta_i(x)\right) = \sum_{i=0}^{2^k-1} \alpha_i W_i(u), \quad (3.14)$$

for all $u \in \mathbb{Z}_2^n$, where

$$W_i(u) = \sum_{x \in \mathbb{Z}_2^n} \Theta_i(x)(-1)^{u \cdot x} = \sum_{x \in \mathbb{Z}_2^n} (-1)^{z_{i,0}a_0(x) \oplus \cdots \oplus z_{i,k-1}a_{k-1}(x) \oplus u \cdot x}, \quad (3.15)$$

i.e., $W_i(u)$ is the WHT of the function $z_{i,0}a_0(x) \oplus \cdots \oplus z_{i,k-1}a_{k-1}(x)$ at point $u \in \mathbb{Z}_2^n$, where $z_i = (z_{i,0}, \ldots, z_{i,k-1}) \in \mathbb{Z}_2^k$, $i = 0, \ldots, 2^k - 1$. Now we state the main result of this section.

Theorem 3.1.9 *Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, $2^{k-1} < q \leq 2^k$, where $f(x)$ is given by (3.25). Let the function $\Theta_i(x)$ be defined by (3.6), and let $W_i(u)$ denote the WHT of the Boolean function $z_{i,0}a_0(x) \oplus \cdots \oplus z_{i,k-1}a_{k-1}(x)$ at point $u \in \mathbb{Z}_2^n$ as in (3.15), for $i = 0, \ldots, 2^k - 1$. Then:*

1. *$\zeta^{f(x)}$ can be represented as a linear combination of the functions $\Theta_i(x)$,*

$$\zeta^{f(x)} = \zeta^{\sum_{i=0}^{k-1} c_i a_i(x)} = \sum_{i=0}^{2^k-1} \alpha_i \Theta_i(x),$$

where α_i are given by

$$\alpha_i = 2^{-k} H_{2^k}^{(i)} B,$$

and the matrix B is given by (3.11).

2. *Consequently, $\mathcal{H}_f(u)$ can be represented as a linear combination of $W_i(u)$, i.e.,*

$$\mathcal{H}_f(u) = \sum_{i=0}^{2^k-1} \alpha_i W_i(u), \quad \forall u \in \mathbb{Z}_2^n. \tag{3.16}$$

For instance, Lemma 3.1.2 is an easy corollary of the above result as illustrated in the following example.

Example 3.1.10 *Let $q = 8 = 2^k$, thus $k = 3$, and consider an arbitrary function $f \in \mathcal{GB}_n^q$ given by $f(x) = a_0(x) + 2a_1(x) + 4a_2(x)$, $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$. Then, the GWHT of f at some arbitrary point $u \in \mathbb{Z}_2^n$ is given by*

$$\mathcal{H}_f(u) = \sum_{x \in \mathbb{Z}_2^n} (-1)^{a_{k-1}(x) \oplus u \cdot x} \zeta^{\sum_{i=0}^{k-2} a_i(x) 2^i} = \sum_{x \in \mathbb{Z}_2^n} (-1)^{a_2(x) \oplus u \cdot x} \zeta^{a_0(x) + 2a_1(x)}.$$

Now we would like to represent $\zeta^{a_0(x)+2a_1(x)}$ as a linear combination of functions $\Theta_0(x) = 1$, $\Theta_1(x) = (-1)^{a_0(x)}$, $\Theta_2(x) = (-1)^{a_1(x)}$ and $\Theta_3(x) = (-1)^{a_0(x)+a_1(x)}$, i.e.,

$$\zeta^{a_0(x)+2a_1(x)} = \alpha_0\Theta_0(x) + \alpha_1\Theta_1(x) + \alpha_2\Theta_2(x) + \alpha_3\Theta_3(x),$$

where the coefficients $\alpha_i \in \mathbb{C}$, $i = 0, 1, 2, 3$. For such coefficients, all of the following equalities must be true:

$$\zeta^{a_0(x)+2a_1(x)} = \begin{cases} 1 = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3, & if \ \ (a_0(x'), a_1(x')) = (0,0) \\ \zeta^1 = \alpha_0 - \alpha_1 + \alpha_2 + \alpha_3, & if \ \ (a_0(x'), a_1(x')) = (1,0) \\ \zeta^2 = \alpha_0 + \alpha_1 - \alpha_2 + \alpha_3, & if \ \ (a_0(x'), a_1(x')) = (0,1) \\ \zeta^3 = \alpha_0 - \alpha_1 - \alpha_2 + \alpha_3, & if \ \ (a_0(x'), a_1(x')) = (1,1) \end{cases},$$

for any input $x' \in \mathbb{Z}_2^n$. By Theorem 3.1.9, we have $\Lambda = 2^{-2} H_{2^2} B$ is given by

$$\Lambda = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = 2^{-2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ \frac{\sqrt{2}}{2} + i\frac{\sqrt{2}}{2} \\ i \\ -\frac{1}{\sqrt{2}} + i\frac{1}{\sqrt{2}} \end{pmatrix} = 2^{-2} \begin{pmatrix} 1 + (1+\sqrt{2})i \\ 1 + (1-\sqrt{2})i \\ 1 + \sqrt{2} - i \\ 1 - \sqrt{2} - i \end{pmatrix}.$$

Using Λ we obtain Lemma 3.1.2, since for every $u \in \mathbb{Z}_2^n$ we have

$$\begin{aligned} \mathcal{H}_f(u) &= 2^{-\frac{n}{2}} \sum_{x \in \mathbb{Z}_2^n} (-1)^{a_2(x) \oplus u \cdot x} \zeta^{a_0(x) + 2a_1(x)} = \alpha_0 2^{-\frac{n}{2}} \sum_{x \in \mathbb{Z}_2^n} (-1)^{a_2(x) \oplus u \cdot x} + \\ &+ \alpha_1 2^{-\frac{n}{2}} \sum_{x \in \mathbb{Z}_2^n} (-1)^{a_0(x) \oplus a_2(x) \oplus u \cdot x} + \alpha_2 2^{-\frac{n}{2}} \sum_{x \in \mathbb{Z}_2^n} (-1)^{a_1(x) \oplus a_2(x) \oplus u \cdot x} \\ &+ \alpha_3 2^{-\frac{n}{2}} \sum_{x \in \mathbb{Z}_2^n} (-1)^{a_0(x) + a_1(x) + a_2(x) \oplus u \cdot x} \\ &= \alpha_0 W_{a_2}(u) + \alpha_1 W_{a_0 + a_2}(u) + \alpha_2 W_{a_1 + a_2}(u) + \alpha_3 W_{a_0 + a_1 + a_2}(u). \end{aligned}$$

Note that in Lemma 3.1.2, the common factor 2^{-2} of the coefficients α_i is moved to the left-hand side by considering $4\mathcal{H}_f(u)$ instead of $\mathcal{H}_f(u)$. Thus, the coefficients α_i above are identical to those in Lemma 3.1.2.

3.2 Sufficient conditions for gbent property (q even)

In this section, we analyze the conditions under which a generalized function $f \in \mathcal{GB}_n^q$ is gbent, where n may be either even and odd. For even q, we provide sufficient conditions for gbent property in terms of the component functions of f. In other words, for this case we give an efficient method for construction of gbent functions using Boolean functions.

Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$ be given in the form (3.25), i.e., $f(x) = \sum_{i=0}^{k-1} a_i(x) 2^i$, and q be even ($2^{k-1} < q \leq 2^k$). For the reasons explained below, we rewrite the function $f(x)$ as

$$f(x) = \frac{q}{2} a(x) + a_0(x) + 2a_1(x) + \ldots + 2^{p-1} a_{p-1}(x), \tag{3.17}$$

for some $p \leq k-1$, where $a, a_i \in \mathcal{B}_n$. We first notice that for $q = 2^k$, by simply taking $p = k-1$, the above form is identical to (3.25) after identifying $a(x) = a_{k-1}(x)$.

The importance of the term $\frac{q}{2} a(x)$ is due to the fact that $\frac{q}{2}$ is the only coefficient from $\mathbb{Z}_q$ for which it holds that $\zeta^{\frac{q}{2} a(x)} = (-1)^{a(x)}$. This coefficient, which naturally appears when $q = 2^k$ as the coefficient of $a_{k-1}(x)$ in (3.25), actually made it possible to express the spectral values of the GWHT of f in terms of certain linear combinations of W_i as given by (3.16). This was essentially achieved through an efficient manipulation of the double summation as it was done when deriving (3.14). However, we still can not prove that f must contain the term $\frac{q}{2} a(x)$ in this explicit form but assuming this form the derivation of the sufficient conditions when $q \neq 2^k$ becomes much easier.

Hence, using $\zeta^{\frac{q}{2}a(x)} = (-1)^{a(x)}$ and applying Theorem 3.1.9-(2) on $\zeta^{a_0(x)+2a_1(x)+\ldots+2^{p-1}a_{p-1}(x)}$, the GWHT at point $u \in \mathbb{Z}_2^n$ is given as:

$$\mathcal{H}_f(u) = \sum_{x\in\mathbb{Z}_2^n} (-1)^{a(x)\oplus u\cdot x} \sum_{i=0}^{2^p-1} \alpha_i \Theta_i(x) = \sum_{i=0}^{2^p-1} \alpha_i W_i(u),$$

using the same approach as when deriving (3.14). Here $W_i(u)$ is WHT at point $u \in \mathbb{Z}_2^n$ of functions $a(x) \oplus z_{i,0}a_0(x) \oplus \cdots \oplus z_{i,p-1}a_{p-1}(x)$, $z_i = (z_{i,0}, \ldots, z_{i,p-1}) \in \mathbb{Z}_2^p$, $i = 0, \ldots, 2^p - 1$.

Let us denote the elements of the i-th Hadamard row $H_{2^p}^{(i)}$ by $h_{i,j}$, $0 \le j, i \le 2^p-1$. Since the form (3.17) will impose the system $H_{2^p}\Lambda = B$, where $B = [b_t]_{t=0}^{2^p-1}$ and $b_t = \zeta^t$, a further calculation of GWHT at point $u \in \mathbb{Z}_2^n$ gives:

$$\begin{aligned}\mathcal{H}_f(u) &= \sum_{i=0}^{2^p-1} \alpha_i W_i(u) = \sum_{i=0}^{2^p-1} \left(2^{-p} \sum_{j=0}^{2^p-1} h_{i,j} b_j \right) W_i(u) \\ &= 2^{-p} \sum_{j=0}^{2^p-1} \left(\sum_{i=0}^{2^p-1} h_{i,j} W_i(u) \right) b_j = 2^{-p} \left(\sum_{j=0}^{2^p-1} S_j \cos\frac{2\pi j}{q} + i \sum_{j=0}^{2^p-1} S_j \sin\frac{2\pi j}{q} \right),\end{aligned}$$

where

$$S_j = \sum_{i=0}^{2^p-1} h_{i,j} W_i(u), \quad j = 0, \ldots, 2^p - 1,\ u \in \mathbb{Z}_2^n. \tag{3.18}$$

Defining the column matrices $W = [W_i]_{i=0}^{2^p-1}$ and $S = [S_j]_{j=0}^{2^p-1}$ we have $S = H_{2^p}W$ which in the matrix form is given as,

$$W = \begin{pmatrix} W_0(u) \\ W_1(u) \\ \vdots \\ W_{2^p-1}(u) \end{pmatrix}_{2^p\times 1}, \qquad S = \begin{pmatrix} S_0 \\ S_1 \\ \vdots \\ S_{2^p-1} \end{pmatrix}_{2^p\times 1} = \begin{pmatrix} H_{2^p}^{(0)}W \\ H_{2^p}^{(1)}W \\ \vdots \\ H_{2^p}^{(2^p-1)}W \end{pmatrix}. \tag{3.19}$$

Consequently, we may write $\mathcal{H}_f(u) = 2^{-p}(S^T B)$, where $B = [\zeta^t]_{t=0}^{2^p-1}$ and S^T is the transpose of S. Note that both the matrix S as well as W depend on the input u, and for every $j = 0, \ldots, 2^p - 1$, we have $S_j = H_{2^p}^{(j)}W$, since H_{2^p} is symmetric. A well-known property of a Hadamard matrix of the size 2^p is that any two distinct rows are orthogonal, thus $\sum_t h_{it}h_{jt} = 0$ for $i \ne j$, and if $i = j$ then $\sum_t h_{it}h_{jt} = 2^p$. The absolute value of $\mathcal{H}_f(u)$ is given as:

$$2^{2p}|\mathcal{H}_f(u)|^2 = \left(\sum_{j=0}^{2^p-1} S_j \cos\frac{2\pi j}{q} \right)^2 + \left(\sum_{j=0}^{2^p-1} S_j \sin\frac{2\pi j}{q} \right)^2. \tag{3.20}$$

It is not difficult to see that (3.20) can be written as

$$2^{2p}|\mathcal{H}_f(u)|^2 = \sum_{j=0}^{2^p-1} S_j^2 + 2 \sum_{j=1}^{2^p-1} \cos\frac{2\pi j}{q} \sum_{i=0}^{2^p-1-j} S_i S_{i+j}. \tag{3.21}$$

Theorem 3.2.1 *Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, where $f(x)$ is given in the form (3.17) and $B = [\zeta^j]_{j=0}^{2^p-1}$. Let $W = [W_i(u)]_{i=0}^{2^p-1}$ be a column matrix (3.27), where $W_i(u)$ denotes the WHT at point $u \in \mathbb{Z}_2^n$ of the Boolean function $a(x) \oplus z_{i,0}a_0(x) \oplus \cdots \oplus z_{i,p-1}a_{p-1}(x)$, $z_i = (z_{i,0}, \ldots, z_{i,p-1}) \in \mathbb{Z}_2^p$, $i = 0, \ldots, 2^p - 1$. Then:*

a) Let n be even and $2^{k-1} < q \leq 2^k$ be even. If all functions $a(x) \oplus z_{i,0}a_0(x) \oplus \ldots \oplus z_{i,p-1}a_{p-1}(x)$ are bent Boolean functions, for every $z_i \in \mathbb{Z}_2^p$, $i = 0, \ldots, 2^p - 1$, and there exists $r \in \{0, 1, \ldots, 2^p - 1\}$ so that the transpose of a matrix W defined by (3.27) is equal to $H_{2^p}^{(r)}$, i.e., $W^T = \pm H_{2^p}^{(r)}$ ($\triangle$), then $f(x)$ is gbent.

b) Let n be odd and $q = 2^{p+1} = 2^k$. If all functions $a(x) \oplus z_{i,0}a_0(x) \oplus \ldots \oplus z_{i,p-1}a_{p-1}(x)$ are semi-bent Boolean functions, for every $z_i \in \mathbb{Z}_2^p$, $i = 0, 1, \ldots, 2^p - 1$, and there exists $r \in \{0, 1, \ldots, 2^p - 1\}$ so that $W^T = (\pm\sqrt{2}H_{2^{p-1}}^{(r)}, \boldsymbol{0}_{2^{p-1}})$ or $W^T = (\boldsymbol{0}_{2^{p-1}}, \pm\sqrt{2}H_{2^{p-1}}^{(r)})$ ($\square$) ($\boldsymbol{0}_{2^{p-1}}$ is the all-zero vector of length 2^{p-1}), then $f(x)$ is gbent.

Proof: *a)* Let n be even, and let us assume that all functions $a(x) \oplus z_{i,0}a_0(x) \oplus \ldots \oplus z_{i,p-1}a_{p-1}(x)$ are bent Boolean functions, for every $z_i \in \mathbb{Z}_2^p$, $i = 0, \ldots, 2^p - 1$. In addition, let us assume that there exists an integer $r \in \{0, 1, \ldots, 2^p - 1\}$ so that $W^T = \pm H_{2^p}^{(r)}$. Then the properties of Hadamard matrices in (3.27) imply the following:

$$S = \begin{pmatrix} H_{2^p}^{(0)} \cdot W^T \\ \vdots \\ H_{2^p}^{(r)} \cdot W^T \\ \vdots \\ H_{2^p}^{(2^p-1)} \cdot W^T \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ H_{2^p}^{(r)} \cdot (\pm H_{2^p}^{(r)}) \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \pm 2^p \\ \vdots \\ 0 \end{pmatrix},$$

and for every i and j ($i \neq j$), it holds that $S_i S_j = 0$. Here we regard $H_{2^p}^{(r)}$ and W^T as vectors, and using the dot product we may write $S_r = H_{2^p}^{(r)} \cdot W^T$. In other words, we use this notation to avoid less precise notation $S_r = H_{2^p}^{(r)} W$. Since in the second sum in (3.21) it is not possible that $S_i = S_{i+j}$, for any $j = 1, \ldots, 2^p - 1$ and $i = 0, \ldots, 2^p - 1 - j$, we get that (3.21) is given as

$$2^{2p}|\mathcal{H}_f(u)|^2 = S_r^2 = 2^{2p},$$

which means that $|\mathcal{H}_f(u)|^2 = 1$, i.e., the function $f(x)$ is gbent.
b) Let n be odd and $q = 2^{p+1}$. The condition that all functions $a(x) \oplus z_{i,0}a_0(x) \oplus \ldots \oplus z_{i,p-1}a_{p-1}(x)$ are semi-bent Boolean functions, for every $z_i \in \mathbb{Z}_2^p$, $i = 0, 1, \ldots, 2^p - 1$, means that $W_i(u) \in \{0, \pm\sqrt{2}\}$. First, note that the definition of the Hadamard matrix implies that there are exactly two rows in H_{2^p} whose first half of its entries are equal to each other (and second halves contain opposite signs). More precisely, for any $r \in \{0, 1, \ldots, 2^{p-1} - 1\}$ and for rows given as

$$H_{2^p}^{(r)} = (H_{2^{p-1}}^{(r)}, H_{2^{p-1}}^{(r)}) \quad \wedge \quad H_{2^p}^{(r+2^{p-1})} = (H_{2^{p-1}}^{(r+2^{p-1})}, -H_{2^{p-1}}^{(r+2^{p-1})})$$

it holds that $H_{2^{p-1}}^{(r)} = H_{2^{p-1}}^{(r+2^{p-1})}$. Therefore, the condition $W^T = (\pm\sqrt{2}H_{2^{p-1}}^{(r)}, \mathbf{0}_{2^{p-1}})$ or $W^T = (\mathbf{0}_{2^{p-1}}, \pm\sqrt{2}H_{2^{p-1}}^{(r)})$ implies $S_r = \pm 2^{p-1}\sqrt{2}$ and $S_{r+2^{p-1}} = \pm 2^{p-1}\sqrt{2}$, which gives:

$$S = \begin{pmatrix} H_{2^p}^{(0)} \cdot W^T \\ \vdots \\ H_{2^p}^{(r)} \cdot W^T \\ \vdots \\ H_{2^p}^{(r+2^{p-1})} \cdot W^T \\ \vdots \\ H_{2^p}^{(2^p-1)} \cdot W^T \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \pm 2^{p-1}\sqrt{2} \\ \vdots \\ \pm 2^{p-1}\sqrt{2} \\ \vdots \\ 0 \end{pmatrix}.$$

Hence, for every $i \in \{0, \ldots, 2^p - 1\} \setminus \{r, r + 2^{p-1}\}$ we have that $S_i = 0$. It is not difficult to see that all $\sum_{i=0}^{2^p-1-j} S_i S_{i+j} = 0$ except for the case when $j = 2^{p-1}$, for which we have $\sum_{i=0}^{2^p-1-2^{p-1}} S_i S_{i+2^{p-1}} = S_r S_{r+2^{p-1}} = 2^{2p-1}$. However, using $q = 2^{p+1}$ in the second sum in (3.21), for $j = 2^{p-1}$ we have the coefficient $\cos\frac{2\pi j}{q} = \cos\frac{\pi 2^p}{2^{p+1}} = \cos\frac{\pi}{2} = 0$, which means that the whole second sum in (3.21) is equal to zero. Note that $j = 2^{p-1}$ does not depend on the integer r in ($\square$), and it is not difficult to see that the only value of q for which $\cos\frac{2\pi j}{q} = 0$ is $q = 2^{p+1}$ (due to a fact that q is an integer). Consequently, in (3.21) we have

$$2^{2p}|\mathcal{H}_f(u)|^2 = S_r^2 + S_{r+2^{p-1}}^2 = 2^{2p-1} + 2^{2p-1} = 2^{2p},$$

i.e., $f(x)$ is gbent. ■

Remark 3.2.2 *Since $2^{k-1} < q \leq 2^k$, it is clear that $p \leq h - 1$ in (3.17). Moreover, the condition $q = 2^{p+1}$ in the second statement in Theorem 3.2.1 actually means that $q = 2^k$, since it is the only power of 2 for which it holds $2^{k-1} < q \leq 2^k$. In the case when n and q are even, the gbent functions always exist (consider $f(x) = \frac{q}{2}a(x)$, $a(x)$ any bent Boolean function). The case when n is odd is much more difficult to handle which is also evident through the nonexistence for certain odd n and certain q, see e.g. [68].*

In what follows we discuss some of the following facts:

- The converse of Theorem 3.2.1 holds for $q = 4$ where the condition ($\triangle$) trivially holds, and the function $f(x)$ is given in the form $f(x) = 2a(x) + a_0(x)$ [109], where n is even.
- When $q = 8$ we have Theorem 3.1.5, where the conditions $(*)$ and $(**)$ are actually equivalent to conditions ($\triangle$) and ($\square$), respectively (see Section 3.2.1).

3.2.1 Equivalent forms of conditions ($\triangle$) and ($\square$)

In this section we present two equivalent forms of the condition ($\triangle$) which are actually imposed by the Hadamard recursion (the same applies on the condition

($\square$)). Let us discuss the form of the condition ($\triangle$) in Theorem 3.2.1, where we consider the function $f(x)$ in the form (3.17). Recall that the condition ($\triangle$) regards W^T and $H_{2^p}^{(r)}$ as vectors (as mentioned in the proof of Theorem 3.2.1). Hence, for the WHT coefficients $W_i(u)$ at point $u \in \mathbb{Z}_2^n$ defined in Theorem 3.2.1 we consider the equality of two vectors given by

$$W^T = (W_0(u), W_1(u), \ldots, W_{2^p-1}(u)) = H_{2^p}^{(r)}.$$

Let $H_{2^p}^{(r)}$ ($k \geq 1$) be an arbitrary row of the Hadamard matrix, i.e., $H_{2^p}^{(r)} = (H_{2^{p-1}}^{(r)}, \pm H_{2^{p-1}}^{(r)})$, where $r \in \{0, 1, \ldots, 2^p - 1\}$. This implies that for every $t = 1, 2, \ldots, p$ and $i = 0, 1, \ldots, 2^{t-1} - 1$, it holds $h_{r,i} = \pm h_{r,i+2^{t-1}}$. This further means that the condition $W^T = \pm H_{2^p}^{(r)}$ is equivalent to a set of equalities

$$W_i(u) = \pm W_{i+2^{t-1}}(u), \quad t = 1, 2, \ldots, p, \quad i = 0, 1, \ldots 2^{t-1} - 1, \tag{3.22}$$

where $u \in \mathbb{Z}_2^n$. For convenience, to see that indices t and i actually represent the Hadamard recursion, let as consider an example when $p = 3$:

1. For $t = 1$ we have that i takes only the value 0 and consequently we have $W_i(u) = W_0 = \pm W_{i+2^{t-1}}(u) = \pm W_1(u)$. Clearly, for any value of $W_0(u) = \pm 1$, we have that the vector (row) $(W_0(u), W_1(u)) = (W_0(u), \pm W_0(u))$ is always equal to some row of the Sylvester-Hadamard matrix $\pm H_2$.

2. For $t = 2$ we have that i takes values 0 and 1. For $i = 0$ we have $W_0(u) = \pm W_2(u)$ and $W_1(u) = \pm W_3(u)$. Note that the signs for both equalities are the same. By the previous step and any value $W_0(u) = \pm 1$, we have that the vector $(W_0(u), W_1(u), W_2(u), W_3(u))$ is always equal to some row of the Sylvester-Hadamard matrix $\pm H_{2^2}$. The same calculation further applies for $t = 3 = p$, where $i = 0, 1, 2$, and we get that $(W_0(u), \ldots, W_7(u))$ is always equal to some row of the Sylvester-Hadamard matrix $\pm H_{2^3}$.

It is important to note here that the signs "$\pm$" in every step always depend on the current value of t. For instance, when we previously had $t = 1$, the sign in front of $W_1(u)$ is fixed for all upcoming values of $t > 1$. For $t = 2$, the signs in front of $W_2(u)$ and $W_3(u)$ are also fixed in the same way, etc.

Equivalently, the relation (3.22) suggests that the condition $W^T = \pm H_{2^p}^{(r)}$ can be written in an equivalent way,

$$\prod_{i=0}^{2^{t-1}-1} W_i(u) = \prod_{i=0}^{2^{t-1}-1} (\pm W_{i+2^{t-1}}(u)), \quad \forall t = 1, 2, \ldots, p. \tag{3.23}$$

It is not difficult to see that the condition $(*)$ $W_0(u)W_3(u) = W_1(u)W_2(u)$ in Theorem 3.1.5 is equivalent to equality (3.23) (where $p = 3$).

In the case when n is even, the discussion above provides some equivalent forms of the condition ($\triangle$). However, in the case when n is odd we have one additional property on Walsh-Hadamard coefficients $W_i(u)$ in the condition ($\square$). First note that condition $W^T = (\pm\sqrt{2}H_{2^p}^{(r)}, \mathbf{0}_{2^p})$ or $W^T = (\mathbf{0}_{2^p}, \pm\sqrt{2}H_{2^p}^{(r)})$, for

some $r \in \{0, 1, \ldots, 2^p - 1\}$, means that we can apply the discussion above on half part of W^T, i.e., on $\pm\sqrt{2}H_{2^p}^{(r)}$. Here we mean that signs of half coordinates of W^T must satisfy the Sylvester-Hadamard recurrence formula. However, for $i = 0, 1, \ldots 2^{p-1} - 1$ we have $W_i(u)W_{2^p-i-1}(u) = 0$ ($t = p$ here), since half coordinates of W^T are zeroes. The equality $W_i(u)W_{2^p-i-1}(u) = 0$, for $i = 0, 1, \ldots 2^{p-1} - 1$, $u \in \mathbb{Z}_2^n$, means that the functions $a(x) \oplus z_{i,0}a_0(x) \oplus z_{i,1}a_1(x) \oplus \ldots \oplus z_{i,p-1}a_{p-1}(x)$ and $a(x) \oplus z_{2^t-i-1,0}a_0(x) \oplus z_{2^t-i-1,1}a_1(x) \oplus \ldots \oplus z_{2^t-i-1,p-1}a_{p-1}(x)$ are disjoint spectra functions [99]. More precisely, in condition ($\square$) we see that for any $i \in \{0, \ldots, 2^{p-1} - 1\}$ and $j \in \{2^{p-1}, \ldots, 2^p - 1\}$ we have that $W_i(u)W_j(u) = 0$, for any $u \in \mathbb{Z}_2^n$.

3.2.2 Necessary and sufficient conditions for the GMMF class

For any arbitrary positive even integer q, an arbitrary gbent function $f : \mathbb{Z}_2^{2n} \to \mathbb{Z}_q$ that belongs to the GMMF class (for instance see [111]) is defined as

$$f(x, y) = \frac{q}{2}x \cdot \sigma(y) + g(y),$$

where σ is a permutation on $\mathbb{Z}_2^n$ and $g : \mathbb{Z}_2^n \to \mathbb{Z}_q$ an arbitrary generalized function from $\mathcal{GB}_n^q$. We see that here $f(x, y)$ contains the term $\frac{q}{2}a(x)$, where $a(x, y) = x \cdot \sigma(y)$, and therefore only $g(y)$ remains to be described in terms of the component Boolean functions by means of Theorem 3.2.1 (due to its connection with $W_i(u)$).

With the following proposition, we prove that all functions from the GMMF class trivially satisfy both conditions in Theorem 3.2.1.

Proposition 3.2.3 *Every gbent function from GMMF class satisfies the converse of Theorem 3.2.1.*

Proof: Let the GMMF function $f : \mathbb{Z}_2^{2n} \to \mathbb{Z}_q$, for even $2^{k-1} < q \leq 2^k$, be written in the form

$$f(x, y) = \frac{q}{2}x \cdot \sigma(y) + g(y) = \frac{q}{2}a(x, y) + a_0(y) + 2a_1(y) + \ldots + 2^{p-1}a_{p-1}(y),$$

where $p \leq k - 1$, $a_i \in \mathcal{B}_{2n}$, $a(x, y) = x \cdot \sigma(y)$, and $g(y)$ is uniquely expressed through a_i as $g(y) = a_0(y) + 2a_1(y) + \ldots + 2^{p-1}a_{p-1}(y)$. Since $f(x, y)$ is written in the form (3.17), according to Theorem 3.2.1 we have that $W_i(u)$ is the WHT of the function

$$a(x, y) \oplus z_{i,0}a_0(y) \oplus \ldots \oplus z_{i,p-1}a_{p-1}(y),$$

where $z_i = (z_{i,0}, \ldots, z_{i,p-1}) \in \mathbb{Z}_2^p$, $i = 0, \ldots, 2^p - 1$, $u \in \mathbb{Z}_2^{2n}$. Clearly, for all $i = 0, 1, \ldots, 2^p - 1$, it holds that $W_i(u) = \pm 1$, for every $u \in \mathbb{Z}_2^n$, since all functions above belong to well known Maiorana-McFarland class of bent Boolean functions. This actually proves the first part of converse of Theorem 3.2.1. It only remains to prove that condition ($\triangle$) holds. By relation (3.22), the condition ($\triangle$) is equivalent to the fact that $W_i(u)W_{i+2^{t-1}}(u)$ takes values ± 1 (Section 3.2.1) for all $t = 1, 2, \ldots, p$ and $i = 0, 1, \ldots, 2^{p-1} - 1$. Let us denote

$$z^{(i)}(y) = z_{i,0}a_0(y) \oplus \ldots \oplus z_{i,p-1}a_{p-1}(y),$$

for $i = 0, 1, \ldots, 2^p - 1$. Now, for every $t = 1, 2, \ldots, p$, $i = 0, 1, \ldots, 2^{t-1} - 1$, and $u = (u_1, u_2) \in \mathbb{Z}_2^n \times \mathbb{Z}_2^n$, we have the following calculation:

$$\begin{aligned} 2^{2n} W_i(u) W_{i+2^{t-1}}(u) &= \sum_{x,y \in \mathbb{Z}_2^n} (-1)^{a(x,y) \oplus z^{(i)}(y) + u \cdot (x,y)} \sum_{x,y \in \mathbb{Z}_2^n} (-1)^{a(x,y) \oplus z^{(i+2^{t-1})}(y) + u \cdot (x,y)} \\ &= \left(\sum_{y \in \mathbb{Z}_2^n} (-1)^{z^{(i)}(y) \oplus u_2 \cdot y} \sum_{x \in \mathbb{Z}_2^n} (-1)^{a(x,y) \oplus u_1 \cdot x} \right) \cdot \\ &\cdot \left(\sum_{y \in \mathbb{Z}_2^n} (-1)^{z^{(i+2^{t-1})}(y) \oplus u_2 \cdot y} \sum_{x \in \mathbb{Z}_2^n} (-1)^{a(x,y) \oplus u_1 \cdot x} \right). \end{aligned}$$

Since $\sum_{x \in \mathbb{Z}_2^n} (-1)^{a(x,y) \oplus u_1 \cdot x} = \sum_{x \in \mathbb{Z}_2^n} (-1)^{x \cdot \sigma(y) \oplus u_1 \cdot x} = 0$, unless $\sigma(y) = u_1$ which happens exactly when $y = \sigma^{-1}(u_1)$. In the case $\sigma(y) = u_1$, then $\sum_{x \in \mathbb{Z}_2^n} (-1)^{x \cdot \sigma(y) \oplus u_1 \cdot x} = 2^n$. It is not difficult to see that for any t, i and $y \in \mathbb{Z}_2^n$, it holds that $z^{(i+2^{t-1})}(y) = z^{(i)}(y) \oplus z^{(2^{t-1})}(y)$. Therefore, we have:

$$\begin{aligned} 2^{2n} W_i(u) W_{i+2^{t-1}}(u) &= (2^n (-1)^{z^{(i)}(y) \oplus u_2 \cdot y}) \cdot (2^n (-1)^{z^{(i+2^{t-1})}(y) \oplus u_2 \cdot y}) \\ &= 2^{2n} (-1)^{z^{(i)}(y) \oplus z^{(i+2^{t-1})}(y)} = 2^{2n} (-1)^{z^{(2^{t-1})}(y)}. \end{aligned}$$

where $y = \sigma^{-1}(u_1)$ is fixed, since $u = (u_1, u_2)$ is fixed. Hence, for every $t = 1, 2, \ldots, p$ and $i = 0, 1, \ldots, 2^{t-1} - 1$, we have that $W_i(u) W_{i+2^{t-1}}(u)$ is constant (with value 1 of -1) which corresponds to selected value of t, i.e., the condition ($\triangle$) is satisfied for every $u \in \mathbb{Z}_2^{2n}$ and arbitrary Boolean functions $a_i \in \mathcal{B}_{2n}$, according to Section 3.2.1 and relation (3.22). Recall that for every (but fixed) value of t we have that the sign of $W_{i+2^{t-1}}(u) = \pm W_i(u)$ is fixed for all $i = 0, 1, \ldots, 2^{t-1} - 1$. ∎

3.2.3 Fulfilling the necessary conditions for gbent property

In this section we discuss methods for satisfying the condition ($\triangle$) (or ($\square$)) from Theorem 3.2.1, where we consider $W^T = \pm H_{2^p}^{(r)}$ for some integer $p \geq 1$ and $r \in \{0, 1, \ldots, 2^p - 1\}$. We discuss certain rather trivial approaches to satisfy these conditions, based on the discussion provided in Section 3.2.1.

In essence, for an arbitrary function $g \in \mathcal{B}_n$, using the equality $W_g(u) = -W_{g \oplus 1}(u)$ we are able to choose the component functions in Theorem 3.2.1 so that the condition ($\triangle$) is satisfied. This actually represents a trivial way to satisfy ($\triangle$), since in that case the equality $W^T = \pm H_{2^k}^{(r)}$ does not depend on $u \in \mathbb{Z}_2^n$. Another possible method employs a linear translate of a function, which gives a simple relationship between the Walsh spectra of the given function and its translate. Indeed, if for some fixed $\alpha \in \mathbb{Z}_2^n$ and $g_1, g_2 \in \mathcal{B}_n$ we have $g_1(x) = g_2(x \oplus \alpha)$, for all $x \in \mathbb{Z}_2^n$, then their Walsh spectra are related through $W_{g_1}(u) = (-1)^{u \cdot \alpha} W_{g_2}(u)$, for all $u \in \mathbb{Z}_2^n$. This equality implies that the condition $W^T = \pm H_{2^p}^{(r)}$ actually depends on $u \in \mathbb{Z}_2^n$, which means that the integer r may change for different $u \in \mathbb{Z}_2^n$.

Example 3.2.4 *In this example we present a trivial method of satisfying the condition ($\triangle$) using the equality $W_g(u) = -W_{g\oplus 1}(u)$, for any $g \in \mathcal{B}_n$. Let $q = 16 = 2^4$ and $f(x) = a_0(x) + 2a_1(x) + 2^2a_2(x) + 2^3a_3(x)$. In this case, we have the matrix $W = [W_i(u)]_{i=0}^{2^3-1}$, where $W_i(u)$ is WHT at point $u \in \mathbb{Z}_2^n$ of the function*

$$a_3(x) \oplus z_{i,0}a_0(x) \oplus z_{i,1}a_1(x) \oplus z_{i,2}a_2(x),$$

$z_i = (z_{i,0}, z_{i,1}, z_{i,2}) \in \mathbb{Z}_2^3$. Hence, the component functions are chosen in the following way:

1. *Let $W_0(u) = W_{a_3}(u)$ and $W_1(u) = W_{a_3\oplus a_0}(u)$ be WHTs of two arbitrary bent functions $a_3(x)$ and $a_3(x) \oplus a_0(x)$, i.e., $W_0(u), W_1(u) = \pm 1$, for any $u \in \mathbb{Z}_2^n$. Assuming that $a_3(x)$ is bent, we may for instance take $a_0 \in \mathcal{A}_n$. Alternatively, we can select $a_3(x)$ and $a_0(x)$ to be component functions of some vectorial bent function.*

2. *Now we must select $a_1(x)$ so that $a_3(x) \oplus a_1(x)$ and $a_3(x) \oplus a_0(x) \oplus a_1(x)$ are bent, satisfying additionally*

$$\{W_0(u), W_1(u)\} = \pm\{W_2(u), W_3(u)\},$$

where $W_2(u) = W_{a_3\oplus a_1}(u)$ and $W_3(u) = W_{a_3\oplus a_0\oplus a_1}(u)$. For instance, if we want to have $\{W_0(u), W_1(u)\} = -\{W_2(u), W_3(u)\}$, then we need to choose the function $a_1(x)$ which satisfies

$$a_3(x) \oplus a_1(x) = a_3(x) \oplus 1 \quad \wedge \quad a_3(x) \oplus a_0(x) \oplus a_1(x) = a_3(x) \oplus a_0(x) \oplus 1.$$

Hence, it must the a case that the function $a_1(x)$ is a constant function equal to 1, i.e., $a_1(x) = 1$ for every $x \in \mathbb{Z}_2^n$. On the other side, selecting $a_1(x) = 0$, for every $x \in \mathbb{Z}_2^n$, implies $\{W_0(u), W_1(u)\} = \{W_2(u), W_3(u)\}$.

3. *Now, the rest of functions are chosen with respect to equality*

$$\{W_0(u), W_1(u), W_2(u), W_3(u)\} = \pm\{W_4(u), W_5(u), W_6(u), W_7(u)\},$$

where $W_4(u) = W_{a_3\oplus a_2}$, $W_5(u) = W_{a_3\oplus a_0\oplus a_2}$, $W_6(u) = W_{a_3\oplus a_1\oplus a_2}$ and $W_7(u) = W_{a_3\oplus a_0\oplus a_1\oplus a_2}$. It is not difficult to see that the sign "+" imposes $a_2(x) = 0$, and the sign "$-$" imposes $a_2(x) = 1$, for every $x \in \mathbb{Z}_2^n$.

Since we started with two arbitrary functions $a_3(x)$ and $a_0(x)$, with first choice "$-$" and second "+", it is not difficult to see that all possible values of $W_{a_3}(u)$ and $W_{a_3\oplus a_0}(u)$, due to a previous choice of the component functions, imply that $W^T \in \{\pm H_{2^3}^{(2)}, \pm H_{2^3}^{(3)}\}$.

The question whether there exists more non-trivial methods to satisfy the condition $W^T = H_{2^p}^{(r)}$ remains open.

Remark 3.2.5 *In the case when n is odd, satisfying the condition ($\square$) is more complicated, since W^T involves Sylvester-Hadamard signs and disjoint spectra functions.*

3.3 Full characterization of generalized bent functions

In Section 3.2 we have provided sufficient conditions under which a function $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$ is gbent (given in form (3.17)), where q is an arbitrary even number. In this section, we further prove the necessity of the conditions from Theorem 3.2.1 for the particular case when q is a power of 2, which is actually the most important case from application point of view. We firstly set some preparatory results regarding the Sylvester-Hadamard matrix.

3.3.1 On the Sylvester-Hadamard matrix

In the following lemma we summarize some properties of the Sylvester-Hadamard matrix (say H_{2^k}), where the first two properties below are recalled from Chapter 2 for self-completeness. The first one follows from the recursive definition of H_{2^k}, the second is the well-known property that each row of H_{2^k} is the evaluation (sequence) of some linear function. The third one may be less well known, hence we provide the proof of this property.

Lemma 3.3.1 *(i) Each row of H_{2^k} is uniquely determined by the signs of the entries at positions 2^s, $s = 0, 1, \ldots, k-1$.*

(ii) Let $z_j = (j_0, j_1, \ldots, j_{k-1}) \in \mathbb{F}_2^k$, where $j = \sum_{i=0}^{k-1} j_i 2^i$, $0 \le j \le 2^k - 1$. Then

$$H_{2^k}^{(r)} = ((-1)^{z_0 \cdot z_r}, (-1)^{z_1 \cdot z_r}, \ldots, (-1)^{z_{2^k-1} \cdot z_r}).$$

(iii) Let $W = (w_0, w_1, \ldots, w_{2^k-1})$, where $w_i = \pm 1$, $0 \le i \le 2^k - 1$. Then $W = \pm H_{2^k}^{(r)}$ for some $r \in \{0, \ldots, 2^k - 1\}$ if and only if for any four distinct integers $j, c, l, v \in \{0, \ldots, 2^k - 1\}$ such that $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$ $(z_j, z_c, z_l, z_v \in \mathbb{F}_2^k)$ we have

$$w_j w_c = w_l w_v. \tag{3.24}$$

Proof: *(iii)* Let $W = (w_0, w_1, \ldots, w_{2^k-1}) = \pm H_{2^k}^{(r)}$ for some (fixed) $r \in \{0, \ldots, 2^k - 1\}$, and let $j, c, l, v \in \{0, \ldots, 2^k - 1\}$ be arbitrary distinct integers such that $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$. By (ii),

$$H_{2^k}^{(r)} = ((-1)^{z_r \cdot z_0}, (-1)^{z_r \cdot z_1}, \ldots, (-1)^{z_r} \cdot z_{2^k-1}).$$

Hence relation (3.24) can be written as

$$(-1)^{z_r \cdot z_j} (-1)^{z_r \cdot z_c} = (-1)^{z_r \cdot z_l} (-1)^{z_r \cdot z_v},$$

or equivalently

$$(-1)^{z_r \cdot (z_j \oplus z_c \oplus z_l \oplus z_v)} = 1,$$

which is satisfied for z_j, z_c, z_l, z_v with $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$.
Suppose conversely that (3.24) holds for all j, c, l, z, v with $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$.

Then, to show that $W = \pm H_{2^k}^{(r)}$ for some r, $0 \le r \le 2^k - 1$, we proceed by induction on k. Trivially it holds for $k = 1$, since $\pm H_2^{(r)}$, $r = 0, 1$, covers all possible combinations for (w_0, w_1). For $k = 2$, we first notice that all solutions of the equality $w_0 w_1 = w_2 w_3$ with $w_i = \pm 1, i = 0, 1, 2, 3$, are the quadruples (w_0, w_1, w_2, w_3) containing an even number of -1s. As it is easy to see, all such quadruples W are of the form $W = (w_0, w_1, \pm(w_0, w_1))$, hence equal to $\pm H_4^{(r)}$ for some $r \in \{0, 1, 2, 3\}$. Before we continue with the induction proof, we also add the argument for $k = 3$. With the above argument applied to the quadruples $(4, 5, 6, 7)$ and $(0, 1, 4, 5)$, we get $(w_6, w_7) = \pm(w_4, w_5)$ and $(w_4, w_5) = \pm(w_0, w_1)$. Consequently,

$$\begin{aligned}(w_0, \ldots, w_7) &= (w_0, w_1, \pm(w_0, w_1), \pm(w_0, w_1, \pm(w_0, w_1))) \\ &= \pm(H_{2^2}^{(d)}, \pm H_{2^2}^{(d)}) = \pm H_{2^3}^{(r)},\end{aligned}$$

for some $0 \le r \le 7$.
Now suppose that the following holds for a tuple $W = (w_0, w_1, \ldots, w_{2^{k-1}-1})$ of length 2^{k-1} with entries in $\{-1, 1\}$: If for all $0 \le j < c < l < v \le 2^{k-1} - 1$ with $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$ we have $w_j w_c = w_l w_v$, then $W = \pm H_{2^{k-1}}^{(r)}$ for some $r \in \{0, 1, \ldots, 2^{k-1} - 1\}$.

Let now $W = (w_0, w_1, \ldots, w_{2^k-1})$, $w_i = \pm 1$, $i = 0, 1, \ldots, 2^k - 1$, such that $w_j w_c = w_l w_v$ for all $0 \le j < c < l < v \le 2^k - 1$ with $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$. By induction hypothesis, we then have $(w_0, w_1, \ldots, w_{2^{k-1}-1}) = \pm H_{2^{k-1}}^{(r)}$ and $(w_{2^{k-1}}, w_{2^{k-1}+1}, \ldots, w_{2^k-1}) = \pm H_{2^{k-1}}^{(\bar{r})}$ for some $r, \bar{r} \in \{0, 1, \ldots, 2^{k-1} - 1\}$. We have to show that $\bar{r} = r$, or equivalently $w_{2^{k-1}+j} = w_j$, $j = 0, 1, \ldots, 2^{k-1} - 1$, or $w_{2^{k-1}+j} = -w_j$, $j = 0, 1, \ldots, 2^{k-1} - 1$. By (i), it is sufficient to show that $w_{2^{k-1}+2^s} = w_{2^s}$, $s = 0, 1, \ldots, k - 2$, or $w_{2^{k-1}+2^s} = -w_{2^s}$, $s = 0, 1, \ldots, k - 2$. We consider the quadruples $(j, c, l, v) = (0, 2^s, 2^{k-1}, 2^{k-1} + 2^s)$, $s = 0, 1, \ldots, k - 2$, for which $z_j \oplus z_c \oplus z_l \oplus z_v = 0$ always holds. Since they satisfy (3.24), either $w_0, w_{2^s}, w_{2^{k-1}}, w_{2^{k-1}+2^s}$ have all the same sign, or exactly two of them are negative. Consequently, if $w_0 = w_{2^{k-1}}$, then we must have $w_{2^s} = w_{2^{k-1}+2^s}$, $s = 0, 1, \ldots, k - 2$, and if $w_0 = -w_{2^{k-1}}$, then $w_{2^s} = -w_{2^{k-1}+2^s}$, $s = 0, 1, \ldots, k - 2$. ∎

In what follows we derive and recall some basic results on gbent functions, which are proved useful in the sequel.

Lemma 3.3.2 *Let $k \ge 3$. Then $\sqrt{2}\zeta_{2^k}^j$ is uniquely represented in $\mathbb{Q}(\zeta_{2^k})$ as*

$$\sqrt{2}\zeta_{2^k}^j = \pm\zeta_{2^k}^{J_1} \pm \zeta_{2^k}^{J_2} \in \mathbb{Q}(\zeta_{2^k}).$$

for some $0 \le J_1 < J_2 \le 2^{k-1} - 1$ with $J_2 - J_1 = 2^{k-2}$.

Proof: W.l.o.g. let $\zeta_{2^3} = \zeta_{2^k}^{2^{k-3}} = (1 + i)/\sqrt{2}$, and hence

$$\sqrt{2}\zeta_{2^k}^j = (\zeta_{2^k}^j + i\zeta_{2^k}^j)/\zeta_{2^k}^{2^{k-3}} = \zeta_{2^k}^{j-2^{k-3}} + \zeta_{2^k}^{2^{k-2}}\zeta_{2^k}^{j-2^{k-3}} = \zeta_{2^k}^{j-2^{k-3}} + \zeta_{2^k}^{j+2^{k-3}}.$$

As $\zeta_{2^k}^j = -\zeta_{2^k}^{j-2^{k-1}}$ we can assume that $0 \le j \le 2^{k-1} - 1$. Again using that

$\zeta_{2^k}^{2^{k-1}} = -1$, we can then write $\sqrt{2}\zeta_{2^k}^j$ as

$$\sqrt{2}\zeta_{2^k}^j = \begin{cases} -\zeta_{2^k}^{j-2^{k-3}+2^{k-1}} + \zeta_{2^k}^{j+2^{k-3}} & \text{if } j - 2^{k-3} < 0, \\ \zeta_{2^k}^{j-2^{k-3}} + \zeta_{2^k}^{j+2^{k-3}} & \text{if } 0 \le j - 2^{k-3} < j + 2^{k-3} < 2^{k-1}, \\ \zeta_{2^k}^{j-2^{k-3}} - \zeta_{2^k}^{j+2^{k-3}-2^{k-1}} & \text{if } j + 2^{k-3} \ge 2^{k-1}. \end{cases}$$

In either case $\sqrt{2}\zeta^j$ is of the form $\pm\zeta_{2^k}^{J_1} \pm \zeta_{2^k}^{J_2}$ for some $0 \le J_1 < J_2 \le 2^{k-1} - 1$ with $J_2 - J_1 = 2^{k-2}$. Since $\{1, \zeta_{2^k}, \ldots, \zeta_{2^k}^{2^{k-1}-1}\}$ is a basis of $\mathbb{Q}(\zeta_{2^k})$, this representation is unique. ∎

Recall that to any generalized Boolean function $f : \mathbb{F}_2^n \to \mathbb{Z}_{2^k}$, we may associate the sequence of Boolean functions $a_j \in \mathcal{B}_n$, $j = 0, 1, \ldots, k-1$, for which

$$f(x) = a_0(x) + 2a_1(x) + 2^2 a_2(x) + \cdots + 2^{k-1}a_{k-1}(x), \ \forall x \in \mathbb{F}_2^n. \tag{3.25}$$

For an integer i, $0 \le i \le 2^{k-1} - 1$, with $i = \sum_{j=0}^{2^{k-1}} i_j 2^j$, $i_j \in \{0, 1\}$, we define the i-th *component function* $g_i \in \mathcal{B}_n$ of f as

$$g_i(x) = a_{k-1}(x) \oplus i_0 a_0(x) \oplus \cdots \oplus i_{k-2}a_{k-2}(x). \tag{3.26}$$

For an element $u \in \mathbb{F}_2^n$, let $\mathcal{W}(u) = (\mathcal{W}_{g_0}(u), \mathcal{W}_{g_1}(u), \ldots, \mathcal{W}_{g_{2^{k-1}-1}}(u))$ and let $\mathbf{S}(u) = (S_0, S_1, \ldots, S_{2^{k-1}-1})$ be the vector defined by

$$\mathbf{S}(u) = \begin{pmatrix} S_0 \\ S_1 \\ \vdots \\ S_{2^{k-1}-1} \end{pmatrix} := H_{2^{k-1}} \begin{pmatrix} \mathcal{W}_{g_0}(u) \\ \mathcal{W}_{g_1}(u) \\ \vdots \\ \mathcal{W}_{g_{2^{k-1}-1}}(u) \end{pmatrix}. \tag{3.27}$$

In Section 3.2 it has been shown that $2^{k-1}\mathcal{H}_f(u)$ can be written in terms of integers S_i and powers of ζ^i. This fact is formalized with the following proposition.

Proposition 3.3.3 *Let $f \in \mathcal{GB}_n^{2^k}$ and $u \in \mathbb{F}_2^n$. Then*

$$2^{k-1}\mathcal{H}_f(u) = (1, \zeta_{2^k}, \ldots, \zeta_{2^k}^{2^{k-1}-1})\mathbf{S}(u) = S_0 + S_1\zeta_{2^k} + \cdots + S_{2^{k-1}-1}\zeta_{2^k}^{2^{k-1}-1}.$$

3.3.2 Necessary and sufficient conditions ($q = 2^k$)

In this section we present necessary and sufficient conditions for the gbentness of functions $f \in \mathcal{GB}_n^{2^k}$ given as in (3.25). We provide an equivalent form of these conditions in terms of certain spectral properties of the component functions of f. In the next section, we will use these conditions to completely characterize gbent functions as algebraic objects, which are shown to possess a lot of structure and to have some interesting properties.

Theorem 3.3.4 *Let $f(x) = a_0(x) + \cdots + 2^{k-2}a_{k-2}(x) + 2^{k-1}a_{k-1}(x) \in \mathcal{GB}_n^{2^k}$, and let $g_i(x) = a_{k-1}(x) \oplus i_0 a_0(x) \oplus i_1 a_1(x) \oplus \cdots \oplus i_{k-2}a_{k-2}(x)$, $0 \le i \le 2^{k-1} - 1$, where $i = \sum_{j=0}^{k-2} i_j 2^j$ and $i_j \in \{0, 1\}$.*

(i) If n is even, then f is gbent if and only if g_i is bent for all $0 \le i \le 2^{k-1}-1$, such that for all $u \in \mathbb{V}_n$,

$$\mathbf{W}(u) = (W_{g_0}(u), W_{g_1}(u), \ldots, W_{g_{2^{k-1}-1}}(u)) = \pm 2^{\frac{n}{2}} H^{(r)}_{2^{k-1}} \tag{3.28}$$

for some r, $0 \le r \le 2^{k-1}-1$, depending on u.

(ii) If n is odd, then f is gbent if and only if g_i is semi-bent for all $0 \le i \le 2^{k-1}-1$, such that for all $u \in \mathbb{V}_n$,

$$\mathbf{W}(u) = (\pm 2^{\frac{n+1}{2}} H^{(r)}_{2^{k-2}}, \boldsymbol{0}_{2^{k-2}}) \quad or \quad \mathbf{W}(u) = (\boldsymbol{0}_{2^{k-2}}, \pm 2^{\frac{n+1}{2}} H^{(r)}_{2^{k-2}}) \tag{3.29}$$

for some r, $0 \le r \le 2^{k-2}-1$, depending on u ($\boldsymbol{0}_{2^{k-2}}$ is the all-zero vector of length 2^{k-2}).

Proof: First we consider the case (i) when n is even. The sufficiency of (3.28) has been shown in Section 3.2, though in a more general context for $f \in \mathcal{GB}^q_n$, where q is an arbitrary even integer. For the sake of completeness we include the proof arguments here. Suppose that (3.28) holds, which also implies that all g_i are bent. By the definition of S_t, $0 \le t \le 2^{k-1}-1$, we then have $S_t = 0$ if $t \ne r$, and $S_r = \pm 2^{n/2} 2^{k-1}$. Proposition 3.3.3 then yields $2^{k-1}\mathcal{H}_h(u) = \pm 2^{n/2} 2^{k-1} \zeta_{2^k}^r$, hence f is gbent.

Now, conversely, suppose that f is gbent. By Proposition 3.3.3, we then have

$$S_0 + S_1\zeta_{2^k} + \cdots + S_{2^{k-1}-1}\zeta_{2^k}^{2^{k-1}-1} = 2^{k-1}\mathcal{H}_f(u) = \pm 2^{k-1} 2^{\frac{n}{2}} \zeta_{2^k}^r$$

for some r, $0 \le r \le 2^{k-1}-1$. Since $\{1, \zeta_{2^k}, \ldots, \zeta_{2^k}^{2^{k-1}-1}\}$ is a basis of $\mathbb{Q}(\zeta_{2^k})$, this implies that $S_t = 0$, $0 \le t \le 2^{k-1}-1$, $t \ne r$, and $S_r = \pm 2^{k-1} 2^{\frac{n}{2}}$. By the invertibility of $H_{2^{k-1}}$, the only solution for $\mathbf{W}(u)$ in the resulting linear system is $\mathbf{W}(u) = \pm 2^{\frac{n}{2}} H^{(r)}_{2^{k-1}}$. Hence (3.28) holds, also implying that all g_i are bent.

For the case (ii), when n is odd, the sufficiency of (3.29) has also been shown in Section 3.2. Again, for the sake of completeness, we include the proof arguments. If (3.29) holds, then by (3.27), for $j \in \{r, r+2^{k-2}\}$ we have $S_j = \pm 2^{k-2} 2^{\frac{n+1}{2}}$, and $S_j = 0$ if $j \ne r, r+2^{k-2}$. Hence, from Proposition 3.3.3, we get

$$\mathcal{H}_f(u) = \pm 2^{\frac{n+1}{2}} \zeta_{2^k}^r \pm 2^{\frac{n+1}{2}} \zeta_{2^k}^{r+2^{k-2}} = 2^{\frac{n+1}{2}} \zeta_{2^k}^r (\pm 1 \pm i) = 2^{\frac{n}{2}} \zeta_{2^k}^r \zeta_8^j,$$

for some $j \in \{1, 3, 5, 7\}$. Therefore, f is gbent.

If conversely f is gbent, then by Proposition 3.3.3 we have

$$S_0 + S_1\zeta_{2^k} + \cdots + S_{2^{k-1}-1}\zeta_{2^k}^{2^{k-1}-1} = 2^{k-1}\mathcal{H}_f(u) = 2^{k-1} 2^{\frac{n-1}{2}} \sqrt{2} \zeta_{2^k}^j,$$

for some $0 \le j \le 2^{k-1}-1$. By Lemma 3.3.2, there exists (a unique) r, $0 \le r \le 2^{k-2}-1$, such that

$$\sqrt{2}\zeta_{2^k}^j = \pm \zeta_{2^k}^r \pm \zeta_{2^k}^{r+2^{k-2}}.$$

Combining the two above relations, we have

$$S_0 + S_1\zeta_{2^k} + \cdots + S_{2^{k-1}-1}\zeta_{2^k}^{2^{k-1}-1} = 2^{k-1} 2^{\frac{n-1}{2}} (\pm \zeta_{2^k}^r \pm \zeta_{2^k}^{r+2^{k-2}}).$$

Therefore, $S_r = \pm 2^{k-2}2^{\frac{n+1}{2}}$, $S_{r+2^{k-2}} = \pm 2^{k-2}2^{\frac{n+1}{2}}$, and $S_t = 0$ for $t \neq r, r+2^{k-2}$, i.e.,

$$\mathbf{S}(u) = \begin{pmatrix} S_0 \\ \vdots \\ S_r \\ \vdots \\ S_{r+2^{k-2}} \\ \vdots \\ S_{2^{k-1}-1} \end{pmatrix} = 2^{k-2}2^{\frac{n+1}{2}} \begin{pmatrix} 0 \\ \vdots \\ (-1)^{e_1} \\ \vdots \\ (-1)^{e_2} \\ \vdots \\ 0 \end{pmatrix}, \quad e_1, e_2 \in \{0,1\}.$$

By the invertibility of $H_{2^{k-1}}$, the linear system (3.27) has a unique solution for all four possibilities of $\mathbf{S}(u)$. As now easily observed, these solutions are $(2^{\frac{n+1}{2}}H^{(r)}_{2^{k-2}}, \mathbf{0}_{2^{k-2}})$, $(-2^{\frac{n+1}{2}}H^{(r)}_{2^{k-2}}, \mathbf{0}_{2^{k-2}})$, $(\mathbf{0}_{2^{k-2}}, 2^{\frac{n+1}{2}}H^{(r)}_{2^{k-2}})$, and $(\mathbf{0}_{2^{k-2}}, -2^{\frac{n+1}{2}}H^{(r)}_{2^{k-2}})$ for $(e_1, e_2) = (1,1), (-1,1), (1,-1)$, and $(-1,-1)$, respectively. ∎

Combining Theorem 3.3.4 and Lemma 3.3.1-(iii) gives a characterization of the gbent property in terms of the Walsh spectral values of the component functions. More precisely, the quadruples (four vectors) of a suitable vector space $\mathbb{F}_2^{k-1}$ which build a 2-dimensional flat specify the component functions whose spectra satisfy certain conditions as described below. In other words, the characterization in Theorem 3.3.4, which relates the spectral values of component functions to the rows of Hadamard matrices, turns out to be equivalent to a particular relation of the Walsh spectral values for the above defined quadruples.

Proposition 3.3.5 *(i) Let n be even, $k \geq 3$, and represent $i = \sum_{j=0}^{k-2} i_j 2^j$ for $0 \leq i \leq 2^{k-1}-1$, with $i_j \in \{0,1\}$. Assume $g_i(x) = a_{k-1}(x) \oplus i_0 a_0(x) \oplus i_1 a_1(x) \oplus \cdots \oplus i_{k-2}a_{k-2}(x)$ are bent functions, for $0 \leq i \leq 2^{k-1}-1$. For $u \in \mathbb{F}_2^n$, the condition in Theorem 3.3.4*

$$\mathbf{W}(u) = (W_{g_0}(u), W_{g_1}(u), \ldots, W_{g_{2^{k-1}-1}}(u)) = \pm 2^{\frac{n}{2}} H^{(r)}_{2^{k-1}} \tag{3.30}$$

holds for some $r \in \{0, \ldots, 2^{k-1}-1\}$, if and only if for any four distinct integers $j, c, l, v \in \{0, \ldots, 2^{k-1}-1\}$ such that $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$, the integers $W_{g_j}(u), W_{g_c}(u), W_{g_l}(u), W_{g_v}(u) \in \{-2^{\frac{n}{2}}, 2^{\frac{n}{2}}\}$ satisfy the equality

$$W_{g_j}(u)W_{g_c}(u) = W_{g_l}(u)W_{g_v}(u). \tag{3.31}$$

(ii) Similarly, when n be odd, let us assume that $g_i(x) = a_{k-1}(x) \oplus i_0 a_0(x) \oplus i_1 a_1(x) \oplus \cdots \oplus i_{k-2}a_{k-2}(x)$ are semi-bent functions, for any $0 \leq i \leq 2^{k-1}-1$. Then,

$$\mathbf{W}(u) = (\pm 2^{\frac{n+1}{2}} H^{(r)}_{2^{k-2}}, \boldsymbol{0}_{2^{k-2}})$$

for some $0 \leq r \leq 2^{k-2}-1$, if and only if $W_{g_j}(u) = 0$ for all $2^{k-2} \leq j \leq 2^{k-1}-1$ and $W_{g_j}(u) \neq 0$ for all $0 \leq j \leq 2^{k-2}-1$ such that for any four distinct integers $j, c, l, v \in \{0, \ldots, 2^{k-2}-1\}$ with $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$, the integers $W_{g_j}(u), W_{g_c}(u), W_{g_l}(u), W_{g_v}(u) \in \{-2^{\frac{n+1}{2}}, 2^{\frac{n+1}{2}}\}$ satisfy the equality

$$W_{g_j}(u)W_{g_c}(u) = W_{g_l}(u)W_{g_v}(u). \tag{3.32}$$

A similar statement is valid for $\mathbf{W}(u) = (\mathbf{0}_{2^{k-2}}, \pm 2^{\frac{n+1}{2}} H^{(r)}_{2^{k-2}})$.

PROOF: The proposition follows from Theorem 3.3.4 and Lemma 3.3.1(iii). ∎

3.3.3 Gbent conditions in terms of affine (semi-)bent spaces

In the previous section we have provided two different characterizations of gbent property, though both are closely related to certain properties of the component functions. The derived conditions essentially also capture the inherent properties of the affine spaces of (semi-)bent functions that correspond to gbent functions. In this section we specify these affine spaces of (semi-)bent functions and also address the affine equivalence of gbent functions in a rigorous manner.

We first develop equivalent gbent conditions in terms of affine bent spaces for even n. In this case, by the definition of the dual g^* of a bent function g, the relation (3.32) in Proposition 3.3.5 is equivalent to

$$(-1)^{g_j^*(u)}(-1)^{g_c^*(u)} = (-1)^{g_l^*(u)}(-1)^{g_v^*(u)},$$

for all $u \in \mathbb{F}_2^n$. Hence $g_j^* \oplus g_c^* \oplus g_l^* \oplus g_v^* = 0$, if j, c, l, v satisfy $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$. Observing that $g_j \oplus g_c \oplus g_l \oplus g_v = 0$ if and only if $z_j \oplus z_c \oplus z_l \oplus z_v = \mathbf{0}$, we obtain the following corollary from Theorem 3.3.4 and Proposition 3.3.5.

Corollary 3.3.6 *A function* $f : \mathbb{F}_2^n \to \mathbb{Z}_{2^k}$, *n even, given as* $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-1}a_{k-1}(x)$ *is gbent if and only if*

$$\mathcal{A} = a_{k-1} \oplus \langle a_0, a_1, \ldots, a_{k-2} \rangle$$

is an affine vector space of bent functions such that for any $h_0, h_1, h_2, h_3 \in \mathcal{A}$ *with* $h_0 \oplus h_1 \oplus h_2 \oplus h_3 = 0$ *we have* $h_0^* \oplus h_1^* \oplus h_2^* \oplus h_3^* = 0$. *Equivalently, if* $h_3 = h_0 \oplus h_1 \oplus h_2$, *then* $h_3^* = h_0^* \oplus h_1^* \oplus h_2^*$.

Corollary 3.3.6 generalizes an observation in [76], where the relations between octal gbent functions and a secondary construction of bent functions proposed by Carlet [16] were investigated. We state the version of this construction [16] given by Mesnager in [80].

Proposition 3.3.7 *[80, Th. 4] Let* g_0, g_1, g_2, g_3 *be bent functions from* $\mathbb{F}_2^n$ *to* $\mathbb{F}_2$ *such that* $g_0 \oplus g_1 \oplus g_2 \oplus g_3 = 0$. *Then the function*

$$g_0 g_1 \oplus g_0 g_2 \oplus g_1 g_2$$

is bent if and only if $g_0^* \oplus g_1^* \oplus g_2^* \oplus g_3^* = 0$, *and its dual is* $g_0^* g_1^* \oplus g_0^* g_2^* \oplus g_1^* g_2^*$.

Combining Corollary 3.3.6 and Proposition 3.3.7 we get interesting alternative conditions for gbent functions in $\mathcal{GB}_n^{2^k}$ when n is even.

Corollary 3.3.8 *Let n be even. A function* $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-1}a_{k-1}(x) \in \mathcal{GB}_n^{2^k}$ *is a gbent function if and only if* $\mathcal{A} = a_{k-1} \oplus \langle a_0, a_1, \ldots, a_{k-2} \rangle$ *is an affine vector space of bent functions such that for every (pairwise distinct)* $g_i, g_j, g_l \in \mathcal{A}$ *the function* $g_i g_j \oplus g_i g_l \oplus g_j g_l$ *is bent.*

Remark 3.3.9 *Note that if the bent functions g_i, g_j, g_l are not pairwise distinct, then $g_ig_j \oplus g_ig_l \oplus g_jg_l$ is trivially bent.*

To address the case when n is odd, we show the following analog of Proposition 3.3.7 for semi-bent functions.

Proposition 3.3.10 *Let g_0, g_1, g_2, g_3 be semi-bent functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ such that $g_0 \oplus g_1 \oplus g_2 \oplus g_3 = 0$. Then, the function*

$$g_0g_1 \oplus g_0g_2 \oplus g_1g_2$$

is semibent if and only if for all $u \in \mathbb{F}_2^n$, $W_{g_i}(u) = 0$ for an even number of $i \in \{0,1,2,3\}$, and if $W_{g_i}(u) \neq 0$ for all $i \in \{0,1,2,3\}$, then

$$W_{g_0}(u)W_{g_1}(u) = W_{g_2}(u)W_{g_3}(u), \tag{3.33}$$

or

$$|\{i : W_{g_i}(u) = 2^{(n+1)/2}\}| = 1, 3, \quad \text{but not} \quad W_{g_0}(u) = W_{g_1}(u) = W_{g_2}(u). \tag{3.34}$$

Proof. By [16, Lemma 1] (see also Proposition 2 in [80]), for (pairwise distinct) Boolean functions g_0, g_1, g_2, g_3 such that $g_0 \oplus g_1 \oplus g_2 \oplus g_3 = 0$, the Walsh-Hadamard transform of $g = g_0g_1 \oplus g_0g_2 \oplus g_1g_2$ satisfies

$$W_g(u) = \frac{1}{2}(W_{g_0}(u) + W_{g_1}(u) + W_{g_2}(u) - W_{g_3}(u))$$

for all $u \in \mathbb{F}_2^n$. The correctness of the proposition follows then easily by checking all possible combinations of $W_{g_i}(u)$, $i \in \{0,1,2,3\}$. Note that (3.33) is equivalent to $W_{g_i}(u) = -2^{(n+1)/2}$ for an even number of $i \in \{0,1,2,3\}$. □

Remark 3.3.11 *If for any $u \in \mathbb{F}_2^n$ for which $W_{g_i}(u) \neq 0$, $i = 0,1,2,3$, the condition (3.33) always applies, then $g = g_ig_j \oplus g_ig_l \oplus g_jg_l$ is semi-bent for any $\{i,j,l\} \subset \{0,1,2,3\}$. If for some of $u \in \mathbb{F}_2^n$ we have (3.34), then this is not true.*

Corollary 3.3.12 *Let n be odd. If $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-1}a_{k-1}(x) \in \mathcal{GB}_n^{2^k}$ is a gbent function, then $\mathcal{A} = a_{k-1} \oplus \langle a_0, a_1, \ldots, a_{k-2}\rangle = a_{k-1} \oplus \mathcal{L}$ is an affine vector space of semi-bent functions such that for every (pairwise distinct) $g_i, g_j, g_l \in \mathcal{A}$ the function $G = g_ig_j \oplus g_ig_l \oplus g_jg_l$ is semi-bent. Moreover, for every $u \in \mathbb{F}_2^n$ we have*

$$\begin{aligned} &W_g(u) = 0 \ \text{ if and only if } \ g \in a_{k-1} \oplus \langle a_0, a_1, \ldots, a_{k-3}\rangle, \ \text{or} \\ &W_g(u) \neq 0 \ \text{ if and only if } \ g \in a_{k-1} \oplus \langle a_0, a_1, \ldots, a_{k-3}\rangle. \end{aligned} \tag{3.35}$$

Conversely, if $\mathcal{A} = a_{k-1} \oplus \mathcal{L}$ is an affine vector space of semi-bent functions such that for every (pairwise distinct) $g_i, g_j, g_l \in \mathcal{A}$ the function $G = g_ig_j \oplus g_ig_l \oplus g_jg_l$ is semi-bent, and $\mathcal{A} = a_{k-1} \oplus \langle a_{k-2}, \mathcal{L}_1\rangle$ for some subspace $\mathcal{L}_1$ of $\mathcal{L}$ and $a_{k-2} \notin \mathcal{L}_1$, with the property that for all $u \in \mathbb{F}_2^n$ we have

$$\begin{aligned} &W_g(u) = 0 \ \text{ if and only if } \ g \in a_{k-1} \oplus \mathcal{L}_1, \ \text{or} \\ &W_g(u) \neq 0 \ \text{ if and only if } \ g \in a_{k-1} \oplus \mathcal{L}_1, \end{aligned} \tag{3.36}$$

then $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-3}a_{k-3}(x) + 2^{k-2}a_{k-2}(x) + 2^{k-1}a_{k-1}(x)$, $a_i \in \mathcal{L}_1$, $0 \leq i \leq k-3$, is gbent.

PROOF: Let $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-1}a_{k-1}(x)$ be a gbent function from $\mathbb{F}_2^n$ to $\mathbb{Z}_{2^k}$, n odd, and for $0 \leq i \leq 2^{k-1}-1$ let $g_i(x) = i_0a_0(x) \oplus i_1a_1(x) \oplus \cdots \oplus i_{k-2}a_{k-2}(x) \oplus a_{k-1}(x)$, where $(i_0, i_1, \ldots, i_{k-2}) = z_i$ is the binary representation of i. By Theorem 3.3.4(ii), g_i is semi-bent for all $0 \leq i \leq 2^{k-1}-1$ and

$$\mathbf{W}(u) = (\pm 2^{\frac{n+1}{2}} H^{(r)}_{2^{k-2}}, \mathbf{0}_{2^{k-2}}) \text{ or } \mathbf{W}(u) = (\mathbf{0}_{2^{k-2}}, \pm 2^{\frac{n+1}{2}} H^{(r)}_{2^{k-2}}) \tag{3.37}$$

for some $0 \leq r \leq 2^{k-2}-1$.

Let $0 \leq i < j < l < v \leq 2^{k-1}-1$ be such that $z_i \oplus z_j \oplus z_l \oplus z_v = \mathbf{0}$, where $z_i = (i_0, i_1, \ldots, i_{k-2}) \in \mathbb{F}_2^{k-1}$ is the binary representation of i. Since $i_{k-2} \oplus j_{k-2} \oplus l_{k-2} \oplus v_{k-2} = 0$, the following situations can then occur:

(i) $0 \leq i, j, l, v \leq 2^{k-2}-1$: In this case, either $W_h(u) = 0$ for all $h \in \{g_i, g_j, g_l, g_v\}$, or $W_h(u) \neq 0$ for all $h \in \{g_i, g_j, g_l, g_v\}$. In the latter case, by Proposition 3.3.5, $W_{g_i}(u)W_{g_j}(u) = W_{g_l}(u)W_{g_v}(u)$. In both cases, by Proposition 3.3.10 the function G is semi-bent.

(ii) $2^{k-2} \leq i, j, l, v \leq 2^{k-1}-1$: The same argument as for (i) applies to this case.

(iii) $0 \leq i, j \leq 2^{k-2}-1$, $2^{k-2} \leq l, v \leq 2^{k-1}-1$: In this case, exactly two of $W_{g_i}(u)$, $W_{g_j}(u)$, $W_{g_l}(u)$ $W_{g_v}(u)$ are zero, hence by Proposition 3.3.10 the function G is semi-bent.

Finally, (3.35) follows directly from (3.37).

To show the converse, we first note that (3.36) implies that exactly 2^{k-2} entries of $W(u)$ are zero, all of them being located either at the first or at the second half of $W(u)$. Since we suppose that $g_ig_j \oplus g_ig_l \oplus g_jg_l$ is semi-bent for *all* (pairwise distinct) $g_i, g_j, g_l \in \mathcal{A}$, by Proposition 3.3.10 and Remark 3.3.11, the nonzero half of $W(u)$ equals to $\pm 2^{\frac{n+1}{2}} H^{(r)}_{2^{k-2}}$ for some $0 \leq r \leq 2^{k-2}-1$. As a consequence, f is gbent by Theorem 3.3.4(ii). ∎

3.3.4 Equivalence of gbent functions

We now give the complete characterization of gbent functions, both for even and odd n, as an algebraic object. Similarly to the case of standard bent functions we discuss the concept of affine equivalence of gbent functions.

As already demonstrated, a gbent function $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-2}a_{k-2}(x) + 2^{k-1}a_{k-1}(x)$ gives rise to $\mathcal{A} = a_{k-1} \oplus \langle a_0, \ldots, a_{k-2} \rangle$ which is an affine space of bent functions (semi-bent functions) with certain properties. Thus, it is natural to investigate its correspondence to apparently similar class of functions, namely to vectorial bent functions. Recall that a vectorial bent function $F : \mathbb{F}_2^n \to \mathbb{F}_2^k$, n even, $k \leq n/2$, is a function

$$F(x) = (a_0(x), a_1(x), \ldots, a_{k-1}(x)), \quad a_i \in \mathcal{B}_n,\ 0 \leq i \leq k-1, \tag{3.38}$$

for which every (nontrivial) component function $i_0a_0 \oplus i_1a_1 \oplus \cdots \oplus i_{k-1}a_{k-1}$, $i_j \in \{0,1\}$, $0 \leq j \leq k-1$, is bent. Equivalently, F is a k-dimensional vector space of bent

functions with a basis $\{a_0, a_1, \ldots, a_{k-1}\}$. Changing the basis, that is, performing a coordinate transformation on $\mathbb{F}_2^k$ does not change the vector space. It is rather the representation in the form (3.38) that changes. In spite of a different appearance, the functions are considered to be the same. Furthermore, it is well known that a coordinate transformation on $\mathbb{F}_2^n$ also results in a vectorial bent function, which is said to be equivalent and is not seen as a different object. For these reasons a discussion about the equivalence of gbent functions seems to be in place.

Let $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-2}a_{k-2}(x) + 2^{k-1}a_{k-1}(x) \in \mathcal{GB}_n^{2^k}$ be a gbent function, $b \in \langle a_0, a_1, \ldots, a_{k-2}\rangle$, and let B be an invertible $(k-1)\times(k-1)$-matrix over $\mathbb{F}_2$. Set $\mathbf{a} = (a_0(x), a_1(x), \ldots, a_{k-2}(x))$ and let $B\mathbf{a}^T = (b_0(x), b_1(x), \ldots, b_{k-2}(x))$. Then,

$$a_{k-1} \oplus \langle a_0, a_1, \ldots, a_{k-2}\rangle \quad \text{and} \quad (a_{k-1} \oplus b) \oplus \langle b_0, b_1, \ldots, b_{k-2}\rangle$$

define the same affine space of bent functions respectively semi-bent functions. In particular, when n is even, the function

$$f_1(x) = b_0(x) + 2b_1(x) + \cdots + 2^{k-2}b_{k-2}(x) + 2^{k-1}(a_{k-1}(x) \oplus b(x))$$

is also a gbent function, describing the same object as f does. One has to be little bit more careful when n is odd, since then the vector space $\mathcal{L} = \langle a_0, a_1, \ldots, a_{k-2}\rangle$ contains a subspace $\mathcal{L}_1$ as described in Corollary 3.3.12. Thus, for our standard representation, when f is of the form (3.25), a_{k-2} has to be chosen from $\mathcal{L} \setminus \mathcal{L}_1$.

As for (vectorial) bent functions one can obtain seemingly new gbent functions from a given one by applying a coordinate transformation on $\mathbb{F}_2^n$. Let $f : \mathbb{F}_2^n \to \mathbb{Z}_{2^k}$ and let A be an invertible $n \times n$-matrix over $\mathbb{F}_2$. Then for $u \in \mathbb{F}_2^n$,

$$\begin{aligned}\mathcal{H}_{f(Ax)}(u) &= \sum_{x\in\mathbb{F}_2^n} \zeta_{2^k}^{f(Ax)}(-1)^{u\cdot x} = \sum_{x\in\mathbb{F}_2^n} \zeta_{2^k}^{f(x)}(-1)^{u\cdot A^{-1}x} \\ &= \sum_{x\in\mathbb{F}_2^n} \zeta_{2^k}^{f(x)}(-1)^{(A^{-1})^T u\cdot x} = \mathcal{H}_f((A^{-1})^T u).\end{aligned}$$

Hence $f(Ax)$ is gbent if and only if f is gbent. Consequently, gbentness is invariant under linear coordinate transformations on $\mathbb{F}_2^n$. From the above discussion, when n is even, we may say that $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-2}a_{k-2}(x) + 2^{k-1}a_{k-1}(x)$ and $f_1(x)$ are equivalent if there exist $A \in GL(n, \mathbb{F}_2)$, $B \in GL(k-1, \mathbb{F}_2)$ and $b \in \langle a_0, a_1, \ldots, a_{k-2}\rangle$, such that

$$f_1(x) = b_0(Ax) + 2b_1(Ax) + \cdots + 2^{k-2}b_{k-2}(Ax) + 2^{k-1}b_{k-1}(Ax)$$

with $(b_0(x), b_1(x), \ldots, b_{k-2}(x)) = B\mathbf{a}^T$ and $b_{k-1} = a_{k-1} \oplus b$. When n is odd, we require that the coordinate transformation induced by B leaves the subspace $\mathcal{L}_1$ invariant.

We notice that the gbent property does not require that $a_0, a_1, \ldots, a_{k-2}$ are linearly independent. Hence, the vector space $\mathcal{L} = \langle a_0, a_1, \ldots, a_{k-2}\rangle$ may not have "full" dimension $k-1$. When n is even, in the extreme case $\dim(\mathcal{L}) = 0$, and $f(x) = 2^{k-1}a_{k-1}(x)$ is a gbent function if a_{k-1} is a bent function. Then the image set of f is two-valued taking the values in $\{0, 2^{k-1}\}$, but certainly one will not

consider a_{k-1} and $2^{k-1}a_{k-1}$ as different objects. In general, it is easily verified that if

$$f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-2}a_{k-2}(x) + 2^{k-1}a_{k-1}(x)$$

is a gbent function in $\mathcal{GB}_n^{2^k}$, then

$$\tilde{f}(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-2}a_{k-2}(x) + 2^{r-1}a_{r-1}(x)$$

is a gbent function in $\mathcal{GB}_n^{2^r}$ for any $r \geq k$. However, this quite artificial lifted version of f with a quite restricted image set, is essentially identified with $f \in \mathcal{GB}_n^{2^k}$.

When n is odd, if

$$f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-2}a_{k-2}(x) + 2^{k-1}a_{k-1}(x)$$

is a gbent function in $\mathcal{GB}_n^{2^k}$, then

$$\tilde{f}(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-3}a_{k-3}(x) + 2^{r-2}a_{r-2}(x) + 2^{r-1}a_{r-1}(x)$$

is also a gbent function in $\mathcal{GB}_n^{2^r}$, for any $r \geq k$. Again, we identify this lifted version $\tilde{f}$ with f.

Let n be even and suppose that the vector space $\langle a_0, a_1, \ldots, a_{r-2}\rangle$ has dimension $k-1$ for some $k \leq r$. Then there exists a matrix $B \in GL(r-1, \mathbb{F}_2)$ such that $B(a_0, a_1, \ldots, a_{r-2})^T = (b_0, b_1, \ldots, b_{k-2}, 0 \ldots, 0)$ for some linearly independent $b_0, b_1, \ldots, b_{k-2}$. Hence

$$\tilde{f}_1(x) = a_0(x) + 2a_1(x) + \cdots + 2^{r-1}a_{r-1}(x)$$

is equivalent to

$$\tilde{f} = b_0(x) + 2b_1(x) + \cdots + 2^{k-2}b_{k-2}(x) + 2^{r-1}a_{r-1}(x),$$

which is the lifted version of

$$f = b_0(x) + 2b_1(x) + \cdots + 2^{k-2}b_{k-2}(x) + 2^{k-1}a_{r-1}(x) \in \mathcal{GB}_n^{2^k}.$$

As a consequence of the above discussion, we can restrict ourselves to gbent functions $f(x) = a_0(x)+2a_1(x)+\cdots+2^{k-2}a_{k-2}(x)+2^{k-1}a_{k-1}(x)$ for which $a_0, a_1, \ldots, a_{k-2}$ are linearly independent. The same argument also applies to the n odd case. The following summary of our discussion is fundamental for the characterization and possibly a classification of gbent functions.

- For a gbent function $f(x) = a_0(x)+2a_1(x)+\cdots+2^{k-2}a_{k-2}(x)+2^{k-1}a_{k-1}(x) \in \mathcal{GB}_n^{2^k}$, the set $\{a_0, a_1, \ldots, a_{k-2}\}$ is always linearly independent (otherwise it reduces to a gbent function in $\mathcal{GB}_n^{2^{k'}}$ for some $k' < k$).
- A gbent function is independent from its representation of the form (3.25) via a basis of $\mathcal{L} = \langle a_0, a_1, \ldots, a_{k-2}\rangle$, and the choice of the coset leader a_{k-1} (for odd n the existence of the distinguished subspace $\mathcal{L}_1$ of $\mathcal{L}$ has to be respected in the representation).

We can now state the main theorems, the characterization of gbent functions in terms of affine (semi-)bent spaces.

Theorem 3.3.13 *Let n be even. A gbent function in $\mathcal{GB}_n^{2^k}$ is a $(k-1)$-dimensional affine vector space $\mathcal{A}$ of bent functions such that for every $g_i, g_j, g_l \in \mathcal{A}$ the function $g_ig_j \oplus g_ig_l \oplus g_jg_l$ is bent.*

Theorem 3.3.14 *Let n be odd. A gbent function in $\mathcal{GB}_n^{2^k}$ is a $(k-1)$-dimensional affine vector space $\mathcal{A} = a_{k-1} \oplus \mathcal{L}$ of semi-bent functions for which $g_ig_j \oplus g_ig_l \oplus g_jg_l$ is semi-bent for every $g_i, g_j, g_l \in \mathcal{A}$, and for all $u \in \mathbb{F}_2^n$ we have*

$$W_g(u) = 0 \text{ if and only if } g \in a_{k-1} \oplus \mathcal{L}_1, \text{ or}$$
$$W_g(u) \neq 0 \text{ if and only if } g \in a_{k-1} \oplus \mathcal{L}_1,$$

for some $(k-2)$-dimensional subspace $\mathcal{L}_1$ of $\mathcal{L}$.

3.3.5 $\mathbb{Z}_q$-bent functions and relative difference sets

In this section, n is always even, $q = 2^k$. We recall that a $\mathbb{Z}_q$-bent function is a function from an n-dimensional vector space $\mathbb{F}_2^n$ over $\mathbb{F}_2$ to $\mathbb{Z}_q$, for which

$$\mathcal{H}_f(\alpha, u) = \sum_{x \in \mathbb{F}_2^n} \zeta_q^{\alpha f(x)} (-1)^{u \cdot x}$$

has absolute value $2^{n/2}$ for every $u \in \mathbb{F}_2^n$ and nonzero $\alpha \in \mathbb{Z}_q = \mathbb{Z}_{2^k}$. Equivalently, a $\mathbb{Z}_q$-bent function given by its graph $D = \{(x, f(x)) : x \in \mathbb{F}_2^n\}$ is a $(2^n, 2^k, 2^n, 2^{n-k})$-relative difference set in $\mathbb{F}_2^n \times \mathbb{Z}_{2^k}$. Clearly, a $\mathbb{Z}_q$-bent function is always gbent. In [71] more general vectorial $\mathbb{Z}_q$-bent functions are considered. We focus on the most interesting case where the co-domain is the cyclic group $\mathbb{Z}_q$.

Proposition 3.3.15 *A function $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-1}a_{k-1}(x) \in \mathcal{GB}_n^{2^k}$, n even, is $\mathbb{Z}_q$-bent if and only if $2^t f(x) = 2^t a_0(x) + 2^{t+1}a_1(x) + \cdots + 2^{k-1}a_{k-t-1}(x) \sim a_0(x) + 2a_1(x) + \cdots + 2^{k-t-1}a_{k-t-1}(x)$ is a gbent function with dimension $k-1-t$ for every $t = 0, 1, \ldots, k-1$.*

Proof: If f is $\mathbb{Z}_q$-bent then $|\mathcal{H}_f^{(2^k)}(2^t, u)| = 2^{n/2}$ for every $u \in \mathbb{F}_2^n$ and $t = 0, 1, \ldots, k-1$ by definition. Conversely, as $\mathbb{Q}(\zeta_1)$ and $\mathbb{Q}(\zeta_2)$ are isomorphic for two primitive roots of unity ζ_1, ζ_2 of the same order, we solely require that $|\mathcal{H}_f^{(2^k)}(2^t, u)| = 2^{n/2}$ for every $u \in \mathbb{F}_2^n$ and $t = 0, 1, \ldots, k-1$. Identifying $2^t f(x)$ with $a_0(x) + 2a_1(x) + \cdots + 2^{k-t-1}a_{k-t-1}(x)$, the proposition follows. ∎

Remark 3.3.16 *As for a $\mathbb{Z}_q$-bent function in $\mathcal{GB}_n^{2^k}$ we require that both $f = a_0(x) + 2a_1(x) + \cdots + 2^{k-1}a_{k-1}(x) \in \mathcal{GB}_n^{2^k}$ and $f_1(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-2}a_{k-2}(x) \in \mathcal{GB}_n^{2^{k-1}}$ are gbent, thus $\langle a_0, a_1, \ldots, a_{k-1} \rangle$ is a vector space of bent functions, i.e., a vectorial bent function.*

We continue with two examples of $\mathbb{Z}_q$-bent functions. For the first example, we employ the fact that $h(x,y) = \mathrm{Tr}_m(x\pi(y))$ is a (Maiorana-McFarland) bent function from $\mathbb{F}_{2^m} \times \mathbb{F}_{2^m}$ to $\mathbb{F}_2$ if and only if π is a permutation of $\mathbb{F}_{2^m}$.

Example 3.3.17 *This example is based on the result in [76, Corollary 3]. Let m be an integer divisible by 4 but not by 5, let $b, c \in \mathbb{F}_{2^m}^*$ with $b^4+b+1 = 0$, and let d be the multiplicative inverse of 11 modulo $2^m - 1$. Then the function $f : \mathbb{F}_{2^m} \times \mathbb{F}_{2^m} \to \mathbb{Z}_{2^3}$*

$$f(x,y) = \mathrm{Tr}_m(c(1+b)y^d x) + 2\mathrm{Tr}_m(c(1+b^{-1})y^d x) + 4\mathrm{Tr}_m(cy^d x)$$

is gbent [76]. Now observe that $\mathrm{Tr}_m(c(1+b^{-1})y^d x)$ and $\mathrm{Tr}_m(c(1+b)y^d x) \oplus \mathrm{Tr}_m(c(1+b^{-1})y^d x)$ are both Maiorana-McFarland bent functions. Hence the function $f_1(x,y) = \mathrm{Tr}_m(c(1+b)y^d x)+2\mathrm{Tr}_m(c(1+b^{-1})y^d x)$ is gbent in $\mathcal{GB}_{2m}^4$ by [109, Theorem 32]. The function $f_2(x,y) = \mathrm{Tr}_m(c(1+b)y^d x)$ is bent, thus formally in $\mathcal{GB}_{2m}^2 = \mathcal{B}_{2m}$. Therefore, by Proposition 3.3.15, $f(x,y)$ is $\mathbb{Z}_8$-bent.

As our second example, we analyse the $\mathbb{Z}_q$-bent function given in Theorem 12 in [71] for $t = 1$. The function is defined via spreads and it is not given in the form (3.25). We start by recalling that a spread of $\mathbb{F}_{2^n}$, $n = 2m$, is a family S of $2^m + 1$ subspaces $U_0, U_1, \ldots, U_{2^m}$ of $\mathbb{F}_{2^n}$, whose pairwise intersection is trivial. The classical example is the regular spread, which for $\mathbb{F}_{2^n} = \mathbb{F}_{2^m} \times \mathbb{F}_{2^m}$ is represented by the family $S = \bigcup_{s\in\mathbb{F}_{2^m}} \{(x, sx) : x \in \mathbb{F}_{2^m}\} \cup \{(0, y) : y \in \mathbb{F}_{2^m}\}$. For the regular spread in $\mathbb{F}_{2^n}$ we can take the family $S = \{\alpha_i \mathbb{F}_{2^m} : i = 1, \ldots, 2^m + 1\}$, where $\{\alpha_i : i = 1, \ldots, 2^m + 1\}$ is a set of representatives of the cosets of the subgroup $\mathbb{F}_{2^m}^*$ of the multiplicative group $\mathbb{F}_{2^n}^*$ (one may take the set of the $(2^m + 1)$-th roots of unity).

Example 3.3.18 *Let $U_0, U_1, \ldots, U_{2^m}$ be the elements of a spread of $\mathbb{F}_{2^n}$, $n = 2m$. We first construct a vectorial bent function F, and thereafter a $\mathbb{Z}_q$-bent function f. We notice that F and f are connected as discussed in Remark 3.3.16.*

Let $\phi : \{1, 2, \ldots, 2^{n/2}\} \to \mathbb{F}_2^k$ be a balanced map, thus any $y \in \mathbb{F}_2^k$ has exactly $2^{n/2-k}$ preimages in the set $\{1, 2, \ldots, 2^{n/2}\}$. Then the function $F : \mathbb{F}_{2^n} \to \mathbb{F}_2^k$ given by

$$F(x) = \begin{cases} \phi(s) & : \ x \in U_s,\ 1 \le s \le 2^m, \text{ and } x \neq 0, \\ 0 & : \ x \in U_0, \end{cases}$$

is a vectorial bent function, see e.g. Theorem 4 in [18]. If $a_i \in \mathcal{B}_n$, $0 \le i \le k-1$, are the coordinate functions of F, i.e. if $F(x) = (a_0(x), a_1(x), \ldots, a_{k-1}(x))$, then F is the vector space of bent functions given as $\langle a_0, a_1, \ldots, a_{k-1}\rangle$.

We now proceed with the construction of the $\mathbb{Z}_q$-bent function given as in [71]. From the balanced map ϕ, we obtain in a natural way a balanced map $\bar{\phi}$ from $\{1, 2, \ldots, 2^{n/2}\}$ to $\mathbb{Z}_{2^k}$ defined as $\bar{\phi}(s) = y_0+2y_1+\cdots+2^{k-1}y_{k-1}$ if $\phi(s) = (y_0, y_1, \ldots, y_{k-1})$. By Theorem 12 in [71], the function

$$f(x) = \begin{cases} \bar{\phi}(s) & : \ x \in U_s,\ 1 \le s \le 2^m, \text{ and } x \neq 0, \\ 0 & : \ x \in U_0, \end{cases}$$

from $\mathbb{F}_{2^n}$ to $\mathbb{Z}_{2^k}$ is $\mathbb{Z}_q$-bent. Then, written in the form (3.25), f is represented as $f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-1}a_{k-1}(x)$, with the Boolean functions a_i, $0 \le i \le k-1$, given as above.

We can change the representation of the vectorial bent function F by changing the basis from $\{a_0, a_1, \ldots, a_{k-1}\}$ to $\{a'_0, a'_1, \ldots, a'_{k-1}\}$. The same vectorial bent function has then the representation $F(x) = \{a'_0(x), a'_1(x), \ldots, a'_{k-1}(x)\}$. This change of the basis implies a modification of ϕ and $\bar{\phi}$, and results also in an alternative formal expression for the $\mathbb{Z}_q$-bent function.

We emphasize that the property of being $\mathbb{Z}_q$-bent is much stronger than the property of being vectorial bent. $\mathbb{Z}_q$-bent functions are very interesting vectorial bent functions since they correspond to two relative difference sets with parameters $(2^n, 2^k, 2^n, 2^{n-k})$: First of all, being vectorial bent, they correspond to the relative difference set $D = \{(x, a_0(x), a_1(x), \ldots, a_{k-1}(x)) \ : \ x \in \mathbb{F}_2^n\}$ in $\mathbb{F}_2^n \times \mathbb{F}_2^k$, and secondly, to the relative difference set $R = \{(x, a_0(x) + 2a_1(x) + \cdots + a_{k-1}(x)) \ : \ x \in \mathbb{F}_2^n\}$ in $\mathbb{F}_2^n \times \mathbb{Z}_{2^k}$. Moreover, further relative difference sets are enclosed in such a vector space of bent functions, the relative difference sets of the bent functions of the form $g_i g_j \oplus g_i g_l \oplus g_j g_l$ for some component functions g_i, g_j, g_l. These bent functions are in general not component functions of the vectorial bent function, hence their relative difference sets are not projections of D. Here we have provided a first systematic description of this class of vectorial bent functions. There are many questions on analysis and construction of such functions which one can investigate. We are convinced that these functions are an interesting target for future research.

3.3.6 The dual and Gray map of gbent functions

In this section we firstly describe the dual f^* of an arbitrary gbent function $f \in \mathcal{GB}_n^{2^k}$. Furthermore, the Gray map of gbent functions is considered.

3.3.7 The dual of a gbent function

For even n we will describe the dual f^* of a gbent function $f \in \mathcal{GB}_n^{2^k}$ via the duals of the component functions of f.

Theorem 3.3.19 *Let n be even and $f \in \mathcal{GB}_n^{2^k}$ be a gbent function given as*

$$f(x) = a_0(x) + 2a_1(x) + \cdots + 2^{k-2}a_{k-2}(x) + 2^{k-1}a_{k-1}(x),$$

for some $a_i \in \mathcal{B}_n$ $(i = 0, \ldots, k-1)$, with component functions g_j, $0 \le j \le 2^{k-1} - 1$. Then the dual $f^* \in \mathcal{GB}_n^{2^k}$ of the function f is given as follows:

$$f^*(x) = b_0(x) + 2b_1(x) + \ldots + 2^{k-2}b_{k-2}(x) + 2^{k-1}b_{k-1}(x), \ \ x \in \mathbb{F}_2^n, \qquad (3.39)$$

where $b_{k-1}(x) = a^*_{k-1}(x)$, $b_j(x) = a^*_{k-1}(x) \oplus (a_{k-1} \oplus a_{2^j})^*(x)$, $j = 0, \ldots, k-2$.

Proof: (i) From Theorem 3.1.9 and the regularity of a gbent function f, we have

$$\mathcal{H}_f(u) = \sum_{i=0}^{2^{k-1}-1} \alpha_i W_{g_i}(u) = 2^{\frac{n}{2}} \sum_{i=0}^{2^{k-1}-1} \alpha_i (-1)^{g_i^*(u)} = 2^{\frac{n}{2}} \zeta_{2^k}^{f^*(u)}.$$

Suppose that $f^*(x) = b_0(x) + 2b_1(x) + \cdots + 2^{k-1}b_{k-1}(x)$ and denote the component functions of f^* by $h_i = b_{k-1} \oplus i_0 b_0 \oplus \ldots i_{k-2} b_{k-2}$, $0 \le i \le 2^{k-1} - 1$ $(i = \sum_{j=0}^{k-2} i_j 2^j)$.

By Theorem 3.1.9,

$$\zeta_{2^k}^{f^*(x)} = \sum_{i=0}^{2^{k-1}-1} \alpha_i (-1)^{h_i(x)}.$$

Combining we get

$$\sum_{i=0}^{2^{k-1}-1} \alpha_i (-1)^{h_i(u)} = \sum_{i=0}^{2^{k-1}-1} \alpha_i (-1)^{g_i^*(u)}.$$

Observing that $\alpha_0, \alpha_1, \ldots, \alpha_{2^{k-1}-1}$ are linearly independent $\mathbb{Q}(\zeta)$ (invertible matrix times $(\zeta_0, \zeta_1, \ldots, \zeta_{2^{k-1}-1})$), we obtain $h_i(x) = g_i^*(x)$, $i = 0, 1, \ldots, 2^{k-1} - 1$ (and all $x \in \mathbb{F}_2^n$). Finally, $b_{k-1} = g_0^* = a_{k-1}^*$, and with $g_{2^j}^* = b_{k-1} \oplus b_j$ and $g_{2^j} = a_{k-1} \oplus a_j$, $j = 1, \ldots, k-2$, we get

$$b_j = a_{k-1}^* \oplus (a_{k-1} \oplus a_j)^*, \; j = 1, \ldots, k-2.$$

■

Theorem 3.3.19 generalizes the results in [76] where a similar conclusion was stated for $k = 2, 3$ only. If n is odd, then the component functions of f are semi-bent, hence the description of the dual of f for n even cannot transfer to n odd in a straightforward manner.

3.3.8 The Gray map of gbent functions

In this section we specify the Gray image of any gbent function by showing that its Gray map is a $(k-1)$-plateaued function if n is even, and a $(k-2)$-plateaued function if n is odd. This again generalizes the existing results given in [70, 109] for $k = 2, 3$ and 4.

Let $f : \mathbb{F}_2^n \to \mathbb{Z}_{2^k}$ be a generalized Boolean function given by (3.25), i.e., by

$$f(x) = a_0(x) + 2a_1(x) + 2^2 a_2(x) + \cdots + 2^{k-1} a_{k-1}(x), \; \forall x \in \mathbb{F}_2^n.$$

The *generalized Gray map* $\psi(f) : \mathcal{GB}_n^{2^k} \to \mathcal{B}_{n+k-1}$ of f is defined by, cf. [15],

$$\psi(f)(x, y_0, \ldots, y_{k-2}) = \bigoplus_{i=0}^{k-2} a_i(x) y_i \oplus a_{k-1}(x). \tag{3.40}$$

We start with the following result.

Lemma 3.3.20 *[70, Lemma 15] Let $n, k-1 \geq 2$ be positive integers and $F : \mathbb{F}_2^n \times \mathbb{F}_2^{k-1} \to \mathbb{F}_2$ be defined by*

$$F(x, y_0, \ldots, y_{k-2}) = a_{k-1}(x) \oplus \bigoplus_{i=0}^{k-2} y_i a_i(x), \quad x \in \mathbb{F}_2^n,$$

where $a_i \in \mathcal{B}_n$, $0 \leq i \leq k-1$. Denote by $A(x)$ the vectorial Boolean function $A = (a_0(x), \ldots, a_{k-2}(x))$ and let $u \in \mathbb{F}_2^n$ and $z_r \in \mathbb{F}_2^{k-1}$. The Walsh-Hadamard transform of F at point $(u, z_r) \in \mathbb{F}_2^n \times \mathbb{F}_2^{k-1}$ is then

$$W_F(u, z_r) = \sum_{z_j \in \mathbb{F}_2^{k-1}} (-1)^{z_j \cdot z_r} W_{a_{k-1} \oplus z_j \cdot A}(u) = H_{2^{k-1}}^{(r)} W^T(u),$$

where $\mathbf{W}(u)$ *is the row vector defined by (3.27), i.e.,* $\mathbf{W}(u) = (W_0(u), \ldots, W_{2^{k-1}-1}(u))$ *and* $W_j(u) = W_{a_{k-1} \oplus z_j \cdot A}(u)$, $j = 0, \ldots, 2^{k-1} - 1$.

We now can show that the Gray map of a gbent function in $\mathcal{GB}_n^{2^k}$ is a certain plateaued function, thus generalizing the results on Gray maps in [111] and [70] which were only given for $q = 4, 8, 16$.

Proposition 3.3.21 *Let* $f \in \mathcal{GB}_n^{2^k}$ *be a gbent function, n even. Then* $\psi(f)$ *is a* $(k-1)$*-plateaued function in* $\mathcal{B}_{n+k-1}$*, thus* $W_{\psi(f)} \in \{0, \pm 2^{n/2+k-1}\}$.

Proof: By Theorem 3.3.4, for any $u \in \mathbb{F}_2^n$ we have $\mathbf{W}(u) = \pm 2^{\frac{n}{2}} H_{2^{k-1}}^{(r)}$ (f is gbent), for some $r \in \{0, \ldots, 2^{k-1} - 1\}$. Then for arbitrary $(u, z_j) \in \mathbb{F}_2^n \times \mathbb{F}_2^{k-1}$, where $z_j \in \mathbb{F}_2^{k-1}$, from Lemma 3.3.20 we have $F = \psi(f)$ and thus

$$\begin{aligned} W_{\psi(f)}(u, z_j) &= H_{2^{k-1}}^{(j)} W^T(u) = H_{2^{k-1}}^{(j)} (\pm 2^{\frac{n}{2}} H_{2^{k-1}}^{(r)})^T = \pm 2^{\frac{n}{2}} H_{2^{k-1}}^{(j)} (H_{2^{k-1}}^{(r)})^T \\ &= \begin{cases} \pm 2^{\frac{n}{2}+k-1}, & r = j \\ 0 & r \neq j \end{cases}, \end{aligned}$$

since $H_{2^{k-1}}^{(j)} (H_{2^{k-1}}^{(r)})^T = \begin{cases} 2^{k-1}, & r = j \\ 0, & r \neq j \end{cases}$, where $H_{2^{k-1}}^{(j)}, H_{2^{k-1}}^{(r)}$ are considered as row vectors. Clearly, for $k \geq 1$ we have $W_{\psi(f)}(u, z_j) \in \{0, \pm 2^{\frac{n}{2}+k-1}\}$, which means that $\psi(f)$ is a $(k-1)$-plateaued function in $\mathcal{B}_{n+k-1}$. ∎

Proposition 3.3.22 *Let* $f \in \mathcal{GB}_n^{2^k}$ *be a gbent function, n odd. Then* $\psi(f)$ *is a* $(k-2)$*-plateaued function in* $\mathcal{B}_{n+k-1}$*, thus* $W_{\psi(f)} \in \{0, \pm 2^{\frac{n+1}{2}+k-2}\}$.

Proof: Recall that for any $u \in \mathbb{F}_2^n$ we have

$$\mathbf{W}(u) = (\pm 2^{\frac{n+1}{2}} H_{2^{k-2}}^{(r)}, \mathbf{0}_{2^{k-2}}) \text{ or } \mathbf{W}(u) = (\mathbf{0}_{2^{k-2}}, \pm 2^{\frac{n+1}{2}} H_{2^{k-2}}^{(r)}),$$

for some $r \in \{0, \ldots, 2^{k-2} - 1\}$. Consequently, for any $(u, z_j) \in \mathbb{F}_2^n \times \mathbb{F}_2^{k-1}$,

$$W_{\psi(f)}(u, z_j) = H_{2^{k-1}}^{(j)} W^T(u) = \begin{cases} \pm 2^{\frac{n+1}{2}+k-2}, & r \in \{j, j + 2^{k-2}\} \\ 0 & r \notin \{j, j + 2^{k-2}\} \end{cases},$$

what completes the proof. ∎

Remark 3.3.23 *Note that Proposition 3.3.21 and Proposition 3.3.22 hold for any even q if f is constructed by [52, Theorem 4.1].*

3.4 Construction methods for generalized bent functions

Apart from the general classes of gbent functions mentioned earlier in this chapter, no general construction method for gbent functions has been proposed until now. In this section, based on the use of the well-known Maiorana-McFarland (MM) class of Boolean functions, we give an explicit construction method of gbent functions, for any even $q > 2$ when n is even and for any q of the form $q = 2^r$ (for $r > 1$) when n is odd (Section 3.4.2). Recall that the GMMF class of functions is defined for an even number of variables. Although, in the case when n is even, our construction method provides functions which belong to the GMMF class, a long-term open problem of providing a generic construction of gbent functions for odd n is solved. The method for odd n employs a large class of disjoint spectra semi-bent functions with certain additional properties which may be useful in other cryptographic applications. Additionally, in Section 3.5 we analyze the class of gbent functions of the form $\frac{q}{2}a(x) + kb(x)$, $k \in \{\frac{q}{4}, \frac{3q}{4}\}$, where we show that almost all constructions of gbent functions for $q \in \{4, 8\}$ (see [108, 107, 111, 113]) belong to this class.

3.4.1 Problem description

An intensive study of gbent functions has recently resulted in their complete characterization when q is a power of 2 (some partial results are also given in [116, 72, 70]). Since the analysis of gbent functions provided in previous section is far more extensive than those given in [116, 72, 70], in this section we will mainly refer to the results given there. Using the approach based on Hadamard matrices, recall that in Section 3.3.3 (or Section 3.3.4) it has been shown that gbent functions from $\mathbb{Z}_2^n$ to $\mathbb{Z}_{2^k}$ in algebraic sense correspond to affine spaces of bent or semi-bent functions with certain properties, when n is even or odd, respectively. The problem of providing generic construction methods of gbent functions is therefore closely related to fulfilling these conditions efficiently. For self-completeness we recall the characterization of gbent functions given in Section 3.3.2 (which can also be found in [116]).

Theorem 3.4.1 *Let $f \in \mathcal{GB}_n^{2^p}$ be given as*

$$f(x) = a_0(x) + \cdots + 2^{p-2}a_{p-2}(x) + 2^{p-1}a_{p-1}(x), \tag{3.41}$$

and let $h_i(x) = a_{p-1}(x) \oplus z_i \cdot (a_0(x), \ldots, a_{p-2}(x))$, $i \in [0, 2^{p-1}-1] = \{0, 1, \ldots, 2^{p-1}-1\}$, where $z_i = (i_0, \ldots, i_{p-2}) \in \mathbb{Z}_2^{p-1}$.

(i) If n is even, then f is gbent if and only if h_i is bent for all $0 \leq i \leq 2^{p-1}-1$, such that for all $u \in \mathbb{Z}_2^n$,

$$\mathbf{W}(u) = (W_{h_0}(u), W_{h_1}(u), \ldots, W_{h_{2^{p-1}-1}}(u)) = \pm 2^{\frac{n}{2}} H_{2^{p-1}}^{(r)} \tag{3.42}$$

for some r, $0 \leq r \leq 2^{p-1}-1$, depending on u.

(ii) If n is odd, then f is gbent if and only if h_i is semi-bent for all $0 \leq i \leq 2^{p-1}-1$, such that for all $u \in \mathbb{Z}_2^n$,

$$\mathbf{W}(u) = (\pm 2^{\frac{n+1}{2}} H^{(r)}_{2^{p-2}}, \mathbf{0}_{2^{p-2}}) \quad or \quad \mathbf{W}(u) = (\mathbf{0}_{2^{p-2}}, \pm 2^{\frac{n+1}{2}} H^{(r)}_{2^{p-2}}) \tag{3.43}$$

for some r, $0 \leq r \leq 2^{p-2}-1$, depending on u ($\mathbf{0}_{2^{p-2}}$ is the all-zero vector of length 2^{p-2}).

Remark 3.4.2 *In Theorem 3.4.1 the condition* (3.42) *(n is even) means that any vector $W(u) = (W_{h_0}(u), \ldots, W_{h_{2^{p-1}-1}}(u))$ must be equal to some row (vector) $H^{(r)}_{2^{p-1}}$ of the Hadamard matrix $H_{2^{p-1}}$ multiplied with $\pm 2^{\frac{n}{2}}$, for all $u \in \mathbb{Z}_2^n$. For odd n, the condition* (3.43) *implies that the first (alternatively the second) half of the vector $W(u)$ is equal to some row of the Hadamard matrix $H_{2^{p-2}}$ multiplied by $\pm 2^{\frac{n+1}{2}}$, whereas the second (alternatively the first) half equals to all-zero vector $\mathbf{0}_{2^{p-2}}$.*

The above result implies that the problem of constructing gbent functions is equivalent to finding an affine space of the coordinate functions $\Lambda = a_{p-1}(x) \oplus \langle a_0(x), \ldots, a_{p-2}(x) \rangle$ (corresponding to $h_i(x)$) which are all bent (or semi-bent if n is odd) functions and in addition satisfying the relation (3.42) (alternatively (3.43) if n is odd). The analysis given in Sections 3.3.3 and 3.3.4 indicates that these properties are not easy to satisfy and a trivial approach is to select most of the coordinate functions to be constant or affinely related to each other. In the extreme case, one may, for even n, specify $a_0(x) = \ldots = a_{p-2}(x) = 0$ so that $\Lambda = a_{p-1}(x)$, thus reducing the dimension of Λ to be zero.

According to Corollary 3.3.8 the relation (3.42), for even n, can be equivalently stated as follows: for any three distinct integers $i, j, k \in \{0, \ldots, 2^{p-1}-1\}$, it must hold that $h_i h_j \oplus h_i h_k \oplus h_j h_k$ is a bent function [1], where $h_i, h_j, h_k \in \Lambda$ and the functions h_l are defined as in Theorem 3.4.1. Then, the fact that $h_i h_j \oplus h_i h_k \oplus h_j h_k$ is bent if and only if $h_i^* \oplus h_j^* \oplus h_k^* = (h_i \oplus h_j \oplus h_k)^*$ [80, Theorem 4] clearly indicates the hardness of the imposed conditions. Indeed, the dual of a sum of bent functions is in general not equal to the sum of duals of these functions, except in the cases when these functions are affinely related to each other (thus $h_i = h_j \oplus g$, where g is an affine function) [12, Proposition 3]. A trivial method for satisfying these conditions, as indicated in Example 3.2.4, is to select certain functions to be constant which then significantly limits the number of choices and consequently the cardinality of $\mathcal{GB}_q^n$ is quite small.

The case n being odd appears to be even harder since apart from finding an affine space Λ of semi-bent functions, the condition (3.43) also implicitly involves the disjoint spectra property. More precisely, for any two integers $i \in [0, 2^{p-2}-1]$ and $j \in [2^{p-2}, 2^{p-1}-1]$ it must hold that $W_{h_i}(u) W_{h_j}(u) = 0$, for any $u \in \mathbb{Z}_2^n$, that is, $h_i = a_{p-1} \oplus z_i \cdot (a_0, \ldots, a_{p-2})$ and $h_j = a_{p-1} \oplus z_j \cdot (a_0, \ldots, a_{p-2})$ are disjoint spectra semi-bent functions. Moreover, as observed in Example 3.2.4, a trivial selection of coordinate semi-bent functions is not possible in this case since specifying some of these coordinate functions to be constant would violate the equality

[1] For shortness of notation we usually drop the variables, thus writing h_i instead of $h_i(x)$

$W_{h_i}(u)W_{h_j}(u) = 0$, which needs to be satisfied for any two integers $i \in [0, 2^{p-2} - 1]$ and $j \in [2^{p-2}, 2^{p-1} - 1]$.

The above discussion demonstrates the hardness of the underlying problem and also motivates the need for some efficient and generic construction methods of gbent functions, which is the main objective of this article. Since the n odd case appears to be more difficult then the n even case, we focus on the construction of semi-bent functions $h_i = a_{p-1} \oplus z_i \cdot (a_0, \ldots, a_{p-2})$, $i \in [0, 2^{p-1} - 1]$, satisfying the condition (3.43) along with the mentioned disjoint spectra property. Even though our proposed construction method for odd n can be easily adopted to cover the n even case, the latter case is just briefly mentioned because the GMMF class provides an efficient and generic construction method.

3.4.2 Construction of gbent functions using MM class

In this section, we describe an efficient method (based on a subtle employment of the MM class) for specifying disjoint spectra semi-bent functions satisfying the gbent conditions given by (3.43).

3.4.3 Disjoint spectra semi-bent functions in the MM class

Since our method utilizes the well-known MM-class of functions, we start with the definition of this class. For $x \in \mathbb{Z}_2^s$ and $y \in \mathbb{Z}_2^v$, let $g : \mathbb{Z}_2^{s+v} \to \mathbb{Z}_2$ be defined as

$$g(x, y) = \phi(x) \cdot y \oplus d(x),$$

where $\phi : \mathbb{Z}_2^s \to \mathbb{Z}_2^v$ and $d \in \mathcal{B}_v$ is an arbitrary function. Then, the function g belongs to the MM-class which can also be represented as a concatenation of affine functions (g is an affine function for any fixed x). It is well-known that if $\phi : \mathbb{Z}_2^s \to \mathbb{Z}_2^v$ is injective then the Walsh spectra of g is three-valued and $W_g(u) \in \{0, \pm 2^v\}$, for any $u \in \mathbb{Z}_2^{v+s}$. In particular, when $n = 2k + 1$ is odd then for $v = k$ and $s = k + 1$ the function g is a semi-bent function.

For our purpose, we are interested in finding a set of semi-bent functions such that certain linear combinations of these have the property of being disjoint spectra semi-bent functions. Therefore, we introduce a useful classification of these functions in terms of disjoint image sets of the mapping ϕ. Let $n = 2k + 1$ be an odd positive integer and $\pi : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ be an arbitrary mapping. We can define $\phi : \mathbb{Z}_2^k \to \mathbb{Z}_2^{k+1}$ so that one coordinate is fixed, where without loss of generality (and to avoid complicated notation) we assume that the first coordinate is fixed so that $\phi_j : \mathbb{Z}_2^k \to \mathbb{Z}_2^{k+1}$, for $j = 0, 1$, is defined as:

$$x \overset{\phi_0}{\mapsto} (0, \pi(x)), \quad x \overset{\phi_1}{\mapsto} (1, \pi(x)), \tag{3.44}$$

where $\pi : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$. Then, if π is a permutation the function

$$g_\pi^{(j)}(x, y) = \phi_j(x) \cdot y \oplus d(x), \quad x \in \mathbb{Z}_2^k, \quad y \in \mathbb{Z}_2^{k+1}, \tag{3.45}$$

is a semi-bent function (since ϕ_j is injective), for $j = 0, 1$. Having defined ϕ_j, $j \in \{0, 1\}$, through the mapping π we now introduce two sets that distinguish the

semi-bent property with respect to π,

$$P_n^{(j)} = \{g_\pi^{(j)} : \mathbb{Z}_2^k \times \mathbb{Z}_2^{k+1} \to \mathbb{Z}_2 \mid d(x) = 0 \text{ and } \pi \text{ is a permutation on } \mathbb{Z}_2^k\}, \quad (3.46)$$

and

$$R_n^{(j)} = \{g_\pi^{(j)} : \mathbb{Z}_2^k \times \mathbb{Z}_2^{k+1} \to \mathbb{Z}_2 \mid d(x) = 0 \text{ and } \pi \text{ is not a permutation on } \mathbb{Z}_2^k\}. \quad (3.47)$$

In the sets $P_n^{(j)}$ and $R_n^{(j)}$ the functions $g_\pi^{(j)}$ are defined by (3.45), where (for simplicity of notation used later) we assign $d(x) = 0$ so that $g_\pi^{(j)} = \phi_j(x) \cdot y$, for $j \in \{0, 1\}$. For more clarity, we illustrate this method in the following example.

Example 3.4.3 *Let us for $n = 2k + 1 = 5$ ($k = 2$) construct a semi-bent function in $P_5^{(1)}$. We define the mapping $\phi_1(x) = (1, \pi(x))$ for $x \in \mathbb{Z}_2^2$ as*

$$\phi_1(00) = (\mathbf{1}, 0, 1),\ \phi_1(10) = (\mathbf{1}, 0, 0),\ \phi_1(01) = (\mathbf{1}, 1, 0),\ \phi_1(11) = (\mathbf{1}, 1, 1),$$

where π is obviously a permutation on $\mathbb{Z}_2^2$. Taking $d(x) = 0$ in (3.45), the four subfunctions (obtained by fixing $x \in \mathbb{Z}_2^2$) are then:

$$g_\pi^{(1)}(0, 0, y) = y_0 \oplus y_2;\ \ g_\pi^{(1)}(1, 0, y) = y_0;\ \ g_\pi^{(1)}(0, 1, y) = y_0 \oplus y_1;\ \ g_\pi^{(1)}(1, 1, y) = y_0 \oplus y_1 \oplus y_2.$$

Thus, the function $g_\pi^{(1)}(x, y) = \phi_1(x) \cdot y$ belongs to the set $P_5^{(1)}$.

However, the signs of Walsh coefficients in linear combinations of the coordinate functions are also of great importance due to the fact that, for any $u \in \mathbb{Z}_2^n$, in relation (3.43) for either the first half of the vector $W(u)$ it holds that

$$(W_{h_0}(u), \ldots, W_{h_{2^{p-2}-1}}(u)) = \pm 2^{\frac{n+1}{2}} H_{2^{p-2}}^{(r)}, \quad r \in [0, 2^{p-2} - 1], \quad (3.48)$$

or alternatively for the second half we have

$$(W_{h_{2^{p-2}}}(u), \ldots, W_{h_{2^{p-1}-1}}(u)) = \pm 2^{\frac{n+1}{2}} H_{2^{p-2}}^{(r)}, \quad r \in [0, 2^{p-2} - 1]. \quad (3.49)$$

The following result is proved useful in determining the signs of non-zero Walsh coefficients for semi-bent functions in $P_n^{(j)}$.

Proposition 3.4.4 *Let $g_\pi^{(j)} = \phi_j(x) \cdot y$, be an arbitrary semi-bent function in $P_n^{(j)}$, where $j \in \{0, 1\}$, $n = 2k + 1$, and ϕ_j is given by (3.44). Then, denoting $\omega_2 \in \mathbb{Z}_2^{k+1}$ by $(t, \omega_2') \in \mathbb{Z}_2 \times \mathbb{Z}_2^k$, for $t \in \{0, 1\}$, we have*

$$W_{g_\pi^{(j)}}(\omega_1, \omega_2) = \begin{cases} (-1)^{\omega_1 \cdot \pi^{-1}(\omega_2')}\, 2^{\frac{n+1}{2}}, & t = j \\ 0, & t \neq j \end{cases}, \quad \forall (\omega_1, \omega_2) \in \mathbb{Z}_2^k \times \mathbb{Z}_2^{k+1}. \quad (3.50)$$

Proof: For any $(\omega_1, \omega_2) \in \mathbb{Z}_2^k \times \mathbb{Z}_2^{k+1}$, the coefficient $W_{g_\pi^{(j)}}(\omega_1, \omega_2)$ can be written as

$$\begin{aligned} W_{g_\pi^{(j)}}(\omega_1, \omega_2) &= \sum_{(x,y) \in \mathbb{Z}_2^k \times \mathbb{Z}_2^{k+1}} (-1)^{g_\pi^{(j)}(x,y) \oplus (x,y) \cdot (\omega_1, \omega_2)} = \sum_{x \in \mathbb{Z}_2^k} (-1)^{x \cdot \omega_1} \sum_{y \in \mathbb{Z}_2^{k+1}} (-1)^{g_\pi^{(j)}(x,y) \oplus y \cdot \omega_2} \\ &= \sum_{x \in \mathbb{Z}_2^k} (-1)^{x \cdot \omega_1} \sum_{y \in \mathbb{Z}_2^{k+1}} (-1)^{(j, \pi(x)) \cdot y \oplus y \cdot \omega_2} = \sum_{x \in \mathbb{Z}_2^k} (-1)^{x \cdot \omega_1} \sum_{y \in \mathbb{Z}_2^{k+1}} (-1)^{((j, \pi(x)) \oplus \omega_2) \cdot y}. \end{aligned}$$

The last sum equals zero for any $x \in \mathbb{Z}_2^k$, unless $(j, \pi(x)) \oplus \omega_2 = 0$ in which case the sum equals $2^{k+1} = 2^{\frac{n+1}{2}}$. Using the fact that π is a permutation, the condition $(j, \pi(x)) \oplus \omega_2 = (j \oplus t, \pi(x) \oplus \omega_2') = \mathbf{0}$ is satisfied for $t = j$ and a unique x given by $x = \pi^{-1}(\omega_2')$. ∎

Remark 3.4.5 *Notice that taking two functions $g_\pi^{(j)}, g_\sigma^{(j)} \in R_n^{(j)}$ so that π, σ are not permutations, we may still have the property that $\pi \oplus \sigma$ is a permutation in which case $g_\pi^{(j)} \oplus g_\sigma^{(j)}$ is a semi-bent function.*

Apart from Proposition 3.4.4, one can easily construct disjoint spectra semi-bent functions as follows.

Proposition 3.4.6 *Let $f_\pi \in P_n^{(j)}$, $j \in \{0, 1\}$, and g_σ belong either to $P_n^{(1)}$ or to $R_n^{(1)}$. If $\pi \oplus \sigma$ is a permutation on $\mathbb{Z}_2^k$, then $f_\pi \oplus g_\sigma$ is a semi-bent function and the functions f_π and $f_\pi \oplus g_\sigma$ are disjoint spectra semi-bent functions.*

Proof: If $\pi \oplus \sigma$ is a permutation on $\mathbb{Z}_2^k$, then clearly functions f_π and $f_\pi \oplus g_\sigma$ are semi-bent functions, since $f_\pi \in P_n^{(j)}$ and $f_\pi \oplus g_\sigma$ is given as

$$f_\sigma(x, y) \oplus g_\pi(x, y) = ((i, \sigma(x)) \oplus (j, \pi(x))) \cdot y = ((i \oplus j, \sigma(x) \oplus \pi(x)) \cdot y,$$

for $i, j \in \{0, 1\}$. Furthermore, if $f_\pi \in P_n^{(j)}$, $j \in \{0, 1\}$, and $g_\sigma \in P_n^{(1)}$ or $g_\sigma \in R_n^{(1)}$, then $f_\pi \oplus g_\sigma \in P_n^{(1 \oplus j)}$. The disjoint spectra property follows trivially from Proposition 3.4.4. ∎

The primary condition in Theorem 3.4.1-(ii) is that the component functions $a_0, \ldots, a_{p-2}, a_{p-1} \in \mathcal{B}_n$ are selected so that $h_i = a_{p-1} \oplus z_i \cdot (a_0, \ldots, a_{p-2})$ is a semi-bent function, for any $i \in [0, 2^{p-1} - 1]$. Especially, when $i = 0$ this implies that a_{p-1} has to be a semi-bent function, hence it can be chosen from the set $P_n^{(j)}$. Recall that the vector $W(u)$ at point $u \in \mathbb{Z}_2^n$ is given as

$$W(u) = (W_{h_0}(u), \ldots, W_{h_{2^{p-2}-1}}(u), W_{h_{2^{p-2}}}(u), \ldots, W_{h_{2^{p-1}-1}}(u)),$$

and accordingly the WHTs of h_i, for $i \in [0, 2^{p-1} - 1]$, constitute the first half of $W(u)$, more precisely $(W_{h_0}(u), \ldots, W_{h_{2^{p-2}-1}}(u))$ which does not involve the function a_{p-2}. Nevertheless, this function cannot be arbitrary chosen (for instance cannot be constant) since its presence in h_j when $j \in [2^{p-2}, 2^{p-1} - 1]$ directly affects the disjoint spectra property through $W_{h_i}(u) W_{h_j}(u) = 0$.

3.4.4 Non-trivial selection of component functions, n odd

We now discuss a suitable selection of the coordinate functions $a_{p-1}, a_0, \ldots, a_{p-2}$ from the sets $P_n^{(j)}$ and/or $R_n^{(j)}$. These sets being closely related to mappings over $\mathbb{Z}_2^k$, to every coordinate function $a_{p-1}, a_0, \ldots, a_{p-2}$ we associate the mappings $\sigma, \tau_0, \ldots, \tau_{k-2} : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ as follows:

$$a_{p-1}(x, y) = (j_{p-1}, \sigma(x)) \cdot y, \;\; a_l(x, y) = (j_l, \tau_l(x)) \cdot y, \quad (x, y) \in \mathbb{Z}_2^k \times \mathbb{Z}_2^{k+1}, \; (3.51)$$

where $j_l \in \{0,1\}$ and $l \in [0, p-2]$. Furthermore, let

$$\pi_i = \sigma \oplus z_i \cdot (\tau_0, \ldots, \tau_{p-2}), \tag{3.52}$$

denote linear combinations of $\sigma, \tau_0, \ldots, \tau_{p-2}$, for $i \in [0, 2^{p-1}-1]$, where $\pi_i : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$.

Henceforth, instead of using the notation h_i, we will use a more precise notation $h_{\pi_i}^{(j)}$ which specifies the function $a_{p-1} \oplus z_i \cdot (a_0, \ldots, a_{p-2})$ with respect to relation (3.51), i.e., the functions $h_{\pi_i}^{(j)} = a_{p-1} \oplus z_i \cdot (a_0, \ldots, a_{p-2})$ are given as

$$h_{\pi_i}^{(j)}(x,y) = (j_{p-1} \oplus z_i \cdot (j_0, \ldots, j_{p-2}), \sigma(x) \oplus z_i \cdot (\tau_0(x), \ldots, \tau_{p-2}(x))) \cdot y = (j, \pi_i(x)) \cdot y,$$

where $(x,y) \in \mathbb{Z}_2^k \times \mathbb{Z}_2^{k+1}$ and $j = j_{p-1} \oplus z_i \cdot (j_0, \ldots, j_{p-2}) \in \{0,1\}$ $(z_i \in \mathbb{Z}_2^{p-1})$.

In order to fulfill the primary condition of Theorem 3.4.1-(ii), i.e., to have an affine space of semi-bent functions $\Lambda = a_{p-1} \oplus z_i \cdot (a_0, \ldots, a_{p-2})$, we will assume that $h_{\pi_i}^{(j)}$ belongs to $P_n^{(j)}$ for all $i \in [0, 2^{p-1}-1]$ $(j \in \{0,1\})$.

Remark 3.4.7 *For arbitrary (fixed) integers $j_0, \ldots, j_{p-1} \in \{0,1\}$, notice that for two different vectors z_i and $z_{i'}$ from $\mathbb{Z}_2^{p-1}$, we may have that $a_{p-1} \oplus z_i \cdot (a_0, \ldots, a_{p-2}) \in P_n^{(j)}$ and $a_{p-1} \oplus z_{i'} \cdot (a_0, \ldots, a_{p-2}) \in P_n^{(j')}$ with $j \neq j'$, since vectors z_i and $z_{i'}$ are directly employed in $j = j_{p-1} \oplus z_i \cdot (j_0, \ldots, j_{p-2})$ and $j' = j_{p-1} \oplus z_{i'} \cdot (j_0, \ldots, j_{p-2})$.*

Recall that in relation (3.43) for any input vector $u \in \mathbb{Z}_2^n$ we have that half of the vector $W(u)$ is a non-zero vector, and the remaining half is equal to the zero vector $\mathbf{0}_{2^{p-2}}$. Therefore, to satisfy further the relation (3.43), Proposition 3.4.4 implies that the integer j in function $h_{\pi_i}^{(j)}$ must be fixed for all $i \in [0, 2^{p-2}-1]$ or for all $i \in [2^{p-2}, 2^{p-1}-1]$ (unlike the case mentioned in Remark 3.4.7), depending on vector $u \in \mathbb{Z}_2^n$. More precisely, let us assume that $j = j_{p-1} \oplus z_i \cdot (j_0, \ldots, j_{p-2}) \in \{0,1\}$ is fixed (the same) in functions $h_{\pi_i}^{(j)} \in P_n^{(j)}$ for all $i \in [0, 2^{p-2}-1]$ (with some $j_0, \ldots, j_{p-1} \in \{0,1\}$). For an arbitrary vector $u = (\omega_1, \omega_2) \in \mathbb{Z}_2^k \times \mathbb{Z}_2^{k+1}$, where $\omega_2 = (t, \omega_2') \in \mathbb{Z}_2^{k+1}$, $t \in \{0,1\}$, Proposition 3.4.4 implies that the first half of the vector $W(u)$ (in relation (3.43)) is given as

$$\begin{aligned} &(W_{h_{\pi_0}^{(j)}}(u), \ldots, W_{h_{\pi_{2^{p-2}-1}}^{(j)}}(u)) = \\ &= \begin{cases} \pm 2^{\frac{n+1}{2}} ((-1)^{\omega_1 \cdot \pi_0^{-1}(\omega_2')}, \ldots, (-1)^{\omega_1 \cdot \pi_{2^{p-2}-1}^{-1}(\omega_2')}), & t = j \\ \mathbf{0}_{2^{p-2}} & t \neq j \end{cases}. \end{aligned} \tag{3.53}$$

On the other hand, fixing $j' = j_{p-1} \oplus z_i \cdot (j_0, \ldots, j_{p-2}) \in \{0,1\}$ for all the remaining indices $i \in [2^{p-2}, 2^{p-1}-1]$, the second half of the vector $W(u)$ is given as

$$\begin{aligned} &(W_{h_{\pi_{2^{p-2}}}^{(j')}}(u), \ldots, W_{h_{\pi_{2^{p-1}-1}}^{(j')}}(u)) = \\ &= \begin{cases} \pm 2^{\frac{n+1}{2}} ((-1)^{\omega_1 \cdot \pi_{2^{p-2}}^{-1}(\omega_2')}, \ldots, (-1)^{\omega_1 \cdot \pi_{2^{p-1}-1}^{-1}(\omega_2')}), & t = j' \\ \mathbf{0}_{2^{p-2}} & t \neq j' \end{cases}. \end{aligned} \tag{3.54}$$

The disjoint spectra property in relation (3.43) is described through equality $W_{h_{\pi_i}^{(j)}}(u) W_{h_{\pi_l}^{(j')}}(u) = 0$, for any two integers $i \in [0, 2^{p-2}-1]$ and $l \in [2^{p-2}, 2^{p-1}-1]$.

Obviously, this property is satisfied in relations (3.53) and (3.54) if and only if it holds that $j' = j \oplus 1$, due to Proposition 3.4.6. However, notice that j' depends on j and the function a_{p-2}, due to the fact that a_{p-2} is present in all functions $h_{\pi_l}^{(j')}$, for $l \in [2^{p-2}, 2^{p-1} - 1]$. In particular, writing the index l as $l = i + 2^{p-2}$ it holds that

$$h_{\pi_l}^{(j')} = h_{\pi_{i+2^{p-2}}}^{(j')} = h_{\pi_i}^{(j)} \oplus a_{p-2}, \ \ \forall i \in [0, 2^{p-2} - 1],$$

due to the lexicographic ordering of $\mathbb{Z}_2^{p-1}$. Hence, the disjoint spectra property is fulfilled if and only if $h_{\pi_i}^{(j)} \in P_n^{(j)}$ for all $i \in [0, 2^{p-2} - 1]$, when j is fixed, and in addition it is necessary to select $a_{p-2} \in P_n^{(j\oplus 1)}$ or $a_{p-2} \in R_n^{(j\oplus 1)}$ so that $h_{\pi_{i+2^{p-2}}}^{(j')} = h_{\pi_i}^{(j)} \oplus a_{p-2}$ belongs to $P_n^{(j\oplus 1)}$ $(j' = j \oplus 1)$, for all $i \in [0, 2^{p-2} - 1]$.

Assuming that the disjoint spectra property is satisfied (through a proper selections of $\sigma, \tau_0, \ldots, \tau_{k-2}$), the condition (3.43) will be fully satisfied if permutations $\pi_0, \ldots, \pi_{2^{p-1}-1}$ (defined by (3.52)) satisfy the relations (3.48) and (3.59). In other words, we need to provide a method of construction of these permutations for which in relations (3.53) and (3.54) it holds that

$$((-1)^{\omega_1 \cdot \pi^{-1}_{0+z\cdot 2^{p-2}}(\omega_2')}, \ldots, (-1)^{\omega_1 \cdot \pi^{-1}_{2^{p-2}-1+z\cdot 2^{p-2}}(\omega_2')}) = \pm H_{2^{p-2}}^{(r_z)}, \tag{3.55}$$

for both $z = 0, 1$ and some $0 \leq r_z \leq 2^{p-2} - 1$. Firstly, with the following result we constrain the choice of permutations π_i satisfying the relations (3.53) and (3.54).

Lemma 3.4.8 *Let $\delta_i : \mathbb{Z}_2^k \to \mathbb{Z}_2$, for $i = 0, \ldots, 2^m - 1$. If for a fixed $x \in \mathbb{Z}_2^k$ the equality*

$$((-1)^{\delta_0(x)}, \ldots, (-1)^{\delta_{2^m-1}(x)}) = \pm H_{2^m}^{(r)},$$

holds for some $r \in \{0, \ldots, 2^m - 1\}$, then there exist $a, b \in \mathbb{Z}_2^m$ so that

$$(\delta_0(x), \ldots, \delta_{2^m-1}(x)) = (a \cdot (z_0 \oplus b), \ldots, a \cdot (z_{2^m-1} \oplus b)). \tag{3.56}$$

Proof: The proof follows from the fact that any row of H_{2^m} corresponds to a linear function $l_a \in \mathcal{B}_m$, say $l_a(z) = a \cdot z$, and the minus sign "$-$" is valid for any b such that $a \cdot b = 1$. ∎

The result below gives a general method for constructing permutations π_i defined by (3.52) for which (3.55) holds for both $z = 0, 1$.

Proposition 3.4.9 *Let the mappings $\sigma, \tau_0, \ldots, \tau_{p-2} : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ used in (3.51) and (3.52) be defined as*

$$\sigma(x) = xS \oplus d, \ \ \tau_c(x) = v^{(c)}, \ \ c \in [0, p-2], \ \ \forall x \in \mathbb{Z}_2^k,$$

where $S \in GL(\mathbb{Z}_2^k)$ is an arbitrary matrix in the group of all invertible $k \times k$ binary matrices and $d, v^{(c)} \in \mathbb{Z}_2^k$ are arbitrary (fixed) vectors. Then, the relation (3.55) holds for both $z = 0, 1$.

PROOF: Let $d, v^{(c)} \in \mathbb{Z}_2^k$ be arbitrary (fixed) vectors and $S \in GL(\mathbb{Z}_2^k)$ be any invertible matrix. Let also $u = (\omega_1, \omega_2) \in \mathbb{Z}_2^k \times \mathbb{Z}_2^{k+1}$ be an arbitrary vector, where $\omega_2 = (t, \omega_2')$ ($t \in \{0,1\}$). W.l.o.g. we only consider the case $z = 0$ in (3.55) (which corresponds to (3.48)), since the same arguments apply to the case $z = 1$ (which corresponds to (3.59)). Equivalently, $z = 0$ means that we are considering the case when $t = j$ (the first equation in (3.53)).

Being linear permutations on $\mathbb{Z}_2^k$, the inverse of $\pi_i(x) = xS \oplus d \oplus z_i \cdot (v^{(0)}, v^{(1)}, \ldots, v^{(p-2)})$ is given as

$$\pi_i^{-1}(x) = (x \oplus d \oplus z_i \cdot (v^{(0)}, v^{(1)}, \ldots, v^{(p-2)}))S^{-1}, \quad \forall i \in [0, 2^{p-1} - 1], \ \ \forall x \in \mathbb{Z}_2^k \tag{3.57}$$

Hence, using (3.57) and denoting by $a = (\omega_1 \cdot v^{(0)} S^{-1}, \ldots, \omega_1 \cdot v^{(p-2)} S^{-1}) \in \mathbb{Z}_2^k$ and $b = \omega_1 \cdot (\omega_2' \oplus d) S^{-1} \in \{0,1\}$, it is not difficult to see that for any $i \in [0, 2^{p-1} - 1]$ the term $\omega_1 \cdot \pi_i^{-1}(\omega_2')$, which occurs in (3.53) and (3.54), for any $\omega_2' \in \mathbb{Z}_2^k$ can be written as

$$\omega_1 \cdot \pi_i^{-1}(\omega_2') = a \cdot z_i \oplus b,$$

Consequently, Lemma 3.4.8 implies that

$$((-1)^{\omega_1 \cdot \pi_0^{-1}(\omega_2')}, \ldots, (-1)^{\omega_1 \cdot \pi_{2^{p-2}-1}^{-1}(\omega_2')}) = (-1)^b((-1)^{a \cdot z_0}, \ldots, (-1)^{a \cdot z_{2^{p-2}-1}}) = \pm H_{2^{p-2}}^{(r)},$$

for some $0 \leq r \leq 2^{p-2} - 1$, which means that relation (3.55) holds for $z = 0$. Using the same arguments, the relation (3.55) also holds for $z = 1$, which completes the proof. ∎

Remark 3.4.10 *One may notice that in Proposition 3.4.9, if $p - 1 > 2^k$ then some mappings $\tau_i = v^{(i)} \in \mathbb{Z}_2^k$ will be the same (assuming p is fixed in (3.41)). However, if $p - 1 \leq 2^k$ then all mappings τ_i can be defined to be pairwise different. Moreover, for $p - 1 \leq k$ the affine space $\Lambda = a_{p-1} \oplus \langle a_0, \ldots, a_{p-2} \rangle$ may have the full dimension $p - 1$ if the vectors $v^{(0)}, \ldots, v^{(p-2)} \in \mathbb{Z}_2^k$ constitute a basis of $\mathbb{Z}_2^k$.*

The results/discussions from this subsection allow us to formalize the generic construction method for gbent functions, which is given with the following steps.

Construction 1: Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_{2^p}$ be defined by (3.25), where $n = 2k + 1$ ($k \in \mathbb{N}$) and $p \geq 2$, and let the coordinate functions $a_0, \ldots, a_{p-1}$ be defined by (3.51). The function f is gbent if its coordinate functions are selected as follows:

(1) Select the corresponding permutations $\sigma, \tau_0, \ldots, \tau_{p-2}$ as defined in Proposition 3.4.9.

(2) With respect to the previous step, set $a_{p-1} \in P_n^{(j)}$ for any $j \in \{0,1\}$, $a_0, \ldots, a_{p-3} \in R_n^{(0)}$ and $a_{p-2} \in R_n^{(1)}$.

Remark 3.4.11 *Note that the first construction step above ensures that $\Lambda = a_{p-1} \oplus \langle a_0, \ldots, a_{p-2} \rangle$ is an affine space of semi-bent functions, for which (3.48) and (3.59) are satisfied. The second step ensures the disjoint spectra property in relation (3.43), thus all functions $a_{p-1} \oplus z_i \cdot (a_0, \ldots, a_{p-2}) \in P_n^{(j)}$ for all $i \in [0, 2^{p-2} - 1]$ and $a_{p-1} \oplus z_l \cdot (a_0, \ldots, a_{p-2}) \in P_n^{(j \oplus 1)}$ for all $l \in [2^{p-2}, 2^{p-1} - 1]$.*

3.4.5 The construction when n is even

In general, our method of constructing gbent functions for n odd, summarized in **Construction 1**, heavily relies on Propositions 3.4.6 and 3.4.9. Nevertheless, assuming that the coordinate functions $a_0, \ldots, a_{p-1}$ (and thus the function f given by (3.41)) are defined on $\mathbb{Z}_2^k \times \mathbb{Z}_2^k$ implies that the n even case can be treated quite similarly. Indeed, considering Proposition 3.4.9 as a method of selecting the coordinate functions $a_0, \ldots, a_{p-1}$, then all functions $h_i = a_{p-1} \oplus z_i \cdot (a_0, \ldots, a_{p-2})$ (now defined on $\mathbb{Z}_2^k \times \mathbb{Z}_2^k$) will belong to the MM-class of bent Boolean functions, since $a_{p-1}(x, y) = \sigma(x) \cdot y$ is a bent function, and $a_c(x, y) = \tau_c(x) \cdot y = v^{(c)} \cdot y$, where $v^{(c)} \in \mathbb{Z}_2^k$ and $c \in [0, p-2]$, are linear functions. The resulting gbent function f, given as

$$f(x, y) = v^{(0)} \cdot y + 2v^{(1)} \cdot y + \ldots + 2^{p-2} v^{(p-2)} \cdot y + 2^{p-1} \sigma(x) \cdot y = g(y) + 2^{p-1} \sigma(x) \cdot y,$$

will belong to the GMMF-class of gbent functions. Note that in Section 3.2.2 it has been shown that all functions within the GMMF-class satisfy the condition (3.42).

3.4.6 Illustrating the construction details - an example

In what follows, we illustrate the use of construction steps in **Construction 1** for providing an example of a gbent function, for odd n. Hence, let us consider a generalized function $f : \mathbb{Z}_2^5 \to \mathbb{Z}_{32}$ ($n = 5 = 2k + 1$, $q = 32$) given as

$$f(x) = a_0(x) + 2a_1(x) + 4a_2(x) + 8a_3(x) + 16a_4(x).$$

Recall that the function f is gbent (for n odd) if and only if the set $\Lambda = a_4 \oplus \langle a_0, \ldots, a_3 \rangle$ is an affine space of semi-bent functions satisfying (3.43) (see Theorem 3.4.1). Since $k = 2$, let $\sigma, \tau_0, \ldots, \tau_3 : \mathbb{Z}_2^2 \to \mathbb{Z}_2^2$ correspond to the component functions $a_4, a_0, \ldots, a_3 \in \mathcal{B}_5$, respectively. Using Proposition 3.4.9, we define these component functions via σ, τ_i so that f is a gbent function, as follows:

$$\begin{aligned} \sigma(x) &= x \oplus (0, 1), \;\; \tau_0(x) = v^{(0)} = (1, 0), \;\; \tau_1(x) = v^{(1)} = (0, 1), \\ \tau_2(x) &= v^{(2)} = (0, 0), \;\; \tau_3(x) = v^{(3)} = (1, 1), \end{aligned}$$

for every $x \in \mathbb{Z}_2^2$. Note that the permutation $\sigma(x) = xS \oplus d$ uses the identity matrix S. Thus we complete the first step of **Construction 1**. Consequently, the coordinate functions are defined as

$$a_4(x, y) = (1, \sigma(x)) \cdot y, \;\; a_i(x, y) = (0, \tau_i(x)) \cdot y, \;\; i = 0, 1, 2,$$

$$a_3(x, y) = (1, \tau_3(x)) \cdot y, \quad (x, y) \in \mathbb{Z}_2^2 \times \mathbb{Z}_2^3.$$

Clearly, we have that $a_4 \oplus z_i \cdot (a_0, \ldots, a_3) \in P_5^{(1)}$ for $i \in [0, 7]$ and $a_4 \oplus z_i \cdot (a_0, \ldots, a_3) \in P_5^{(0)}$ for $i \in [8, 15]$, $z_i \in \mathbb{Z}_2^4$, thus satisfying the disjoint spectra property (the choice of a_i is in accordance to the second step in **Construction 1**). Denoting $W_{h_{\pi_i}}(u) = W_{a_4 \oplus z_i \cdot (a_0, \ldots, a_3)}(u)$, for $u \in \mathbb{Z}_2^5$, the vectors $W(u) = (W_{h_{\pi_0}}(u), \ldots, W_{h_{\pi_{15}}}(u))$ are given in Table 3.1. Consequently, the output values of the gbent function f are given by

$$\{0, 0, 0, 0, 24, 24, 24, 24, 9, 25, 9, 25, 17, 1, 17, 1, 26, 26, 10, 10, 2, 2, 18, 18, 19, 3, 3, 19, 11, 27, 27, 11\}.$$

Table 3.1: Vectors $W(u)$ for all $u \in \mathbb{Z}_2^5$.

$u \in \mathbb{Z}_2^5$	$W(u) = (W_{h_{\pi_0}}(u), \ldots, W_{h_{\pi_{15}}}(u))$	$W(u) = \{\mathbf{0}_{2^3}, \pm 8H_{2^3}^{(r)}\}$ or $W^T = \{\pm 8H_{2^3}^{(r)}, \mathbf{0}_{2^3}\}$
u_0	$\{0,0,0,0,0,0,0,0,8,8,8,8,8,8,8,8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(0)}\}$
u_1	$\{0,0,0,0,0,0,0,0,-8,8,-8,8,-8,8,-8,8\}$	$\{\mathbf{0}_{2^3},\ -8H_{2^3}^{(1)}\}$
u_2	$\{0,0,0,0,0,0,0,0,8,8,-8,-8,8,8,-8,-8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(2)}\}$
u_3	$\{0,0,0,0,0,0,0,0,-8,8,8,-8,-8,8,8,-8\}$	$\{\mathbf{0}_{2^3},\ -8H_{2^3}^{(3)}\}$
u_4	$\{8,8,8,8,8,8,8,8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(0)},\ \mathbf{0}_{2^3}\}$
u_5	$\{8,-8,8,-8,8,-8,8,-8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(1)},\ \mathbf{0}_{2^3}\}$
u_6	$\{-8,-8,8,8,-8,-8,8,8,0,0,0,0,0,0,0,0\}$	$\{-8H_{2^3}^{(2)},\ \mathbf{0}_{2^3}\}$
u_7	$\{-8,8,8,-8,-8,8,8,-8,0,0,0,0,0,0,0,0\}$	$\{-8H_{2^3}^{(3)},\ \mathbf{0}_{2^3}\}$
u_8	$\{0,0,0,0,0,0,0,0,8,8,8,8,8,8,8,8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(0)}\}$
u_9	$\{0,0,0,0,0,0,0,0,8,-8,8,-8,8,-8,8,-8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(1)}\}$
u_{10}	$\{0,0,0,0,0,0,0,0,8,8,-8,-8,8,8,-8,-8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(2)}\}$
u_{11}	$\{0,0,0,0,0,0,0,0,8,-8,-8,8,8,-8,-8,8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(3)}\}$
u_{12}	$\{8,8,8,8,8,8,8,8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(0)},\ \mathbf{0}_{2^3}\}$
u_{13}	$\{-8,8,-8,8,-8,8,-8,8,0,0,0,0,0,0,0,0\}$	$\{-8H_{2^3}^{(1)},\ \mathbf{0}_{2^3}\}$
u_{14}	$\{-8,-8,8,8,-8,-8,8,8,0,0,0,0,0,0,0,0\}$	$\{-8H_{2^3}^{(2)},\ \mathbf{0}_{2^3}\}$
u_{15}	$\{8,-8,-8,8,8,-8,-8,8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(3)},\ \mathbf{0}_{2^3}\}$
u_{16}	$\{0,0,0,0,0,0,0,0,8,8,8,8,8,8,8,8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(0)}\}$
u_{17}	$\{0,0,0,0,0,0,0,0,-8,8,-8,8,-8,8,-8,8\}$	$\{\mathbf{0}_{2^3},\ -8H_{2^3}^{(1)}\}$
u_{18}	$\{0,0,0,0,0,0,0,0,-8,-8,8,8,-8,-8,8,8\}$	$\{\mathbf{0}_{2^3},\ -8H_{2^3}^{(2)}\}$
u_{19}	$\{0,0,0,0,0,0,0,0,8,-8,-8,8,8,-8,-8,8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(3)}\}$
u_{20}	$\{8,8,8,8,8,8,8,8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(0)},\ \mathbf{0}_{2^3}\}$
u_{21}	$\{8,-8,8,-8,8,-8,8,-8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(1)},\ \mathbf{0}_{2^3}\}$
u_{22}	$\{8,8,-8,-8,8,8,-8,-8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(2)},\ \mathbf{0}_{2^3}\}$
u_{23}	$\{8,-8,-8,8,8,-8,-8,8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(3)},\ \mathbf{0}_{2^3}\}$
u_{24}	$\{0,0,0,0,0,0,0,0,8,8,8,8,8,8,8,8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(0)}\}$
u_{25}	$\{0,0,0,0,0,0,0,0,8,-8,8,-8,8,-8,8,-8\}$	$\{\mathbf{0}_{2^3},\ 8H_{2^3}^{(1)}\}$
u_{26}	$\{0,0,0,0,0,0,0,0,-8,-8,8,8,-8,-8,8,8\}$	$\{\mathbf{0}_{2^3},\ -8H_{2^3}^{(2)}\}$
u_{27}	$\{0,0,0,0,0,0,0,0,-8,8,8,-8,-8,8,8,-8\}$	$\{\mathbf{0}_{2^3},\ -8H_{2^3}^{(3)}\}$
u_{28}	$\{8,8,8,8,8,8,8,8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(0)},\ \mathbf{0}_{2^3}\}$
u_{29}	$\{-8,8,-8,8,-8,8,-8,8,0,0,0,0,0,0,0,0\}$	$\{-8H_{2^3}^{(1)},\ \mathbf{0}_{2^3}\}$
u_{30}	$\{8,8,-8,-8,8,8,-8,-8,0,0,0,0,0,0,0,0\}$	$\{8H_{2^3}^{(2)},\ \mathbf{0}_{2^3}\}$
u_{31}	$\{-8,8,8,-8,-8,8,8,-8,0,0,0,0,0,0,0,0\}$	$\{-8H_{2^3}^{(3)},\ \mathbf{0}_{2^3}\}$

3.5 Generalized bent functions constructed out of two (generalised) Boolean functions

In this section, we give some necessary and sufficient conditions for generalized bent functions when represented as a linear combination of functions: from $\mathcal{GB}_q^m$ (m even and odd), functions from $\mathcal{B}_n$, and from both $\mathcal{GB}_q^m$ or $\mathcal{B}_n$, thus varying the variable, domain and codomain space. In addition, we consider gbent functions whose restrictions are equal to some linear combination of functions from $\mathcal{B}_n$ and $\mathcal{GB}_q^n$, see Sections 3.5.5 and 3.5.6.

Let $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, $q \geq 2$ and n even. In what follows, we discuss generalized

Boolean functions of the form

$$f(x) = c_1 a(x) + c_2 b(x),$$

where $a, b \in \mathcal{B}_n$, and $c_1, c_2 \in \mathbb{Z}_q$. We want to investigate, under which conditions on the functions a, b and constants c_i, $i = 1, 2$, f is possibly a gbent function. Let $u \in \mathbb{Z}_2^n$ be an arbitrary element. We start with computing the GWHT:

$$2^{\frac{n}{2}} \mathcal{H}_f(u) = \sum_{x \in \mathbb{Z}_2^n} \zeta^{c_1 a(x) + c_2 b(x)} (-1)^{u \cdot x} = \sum_{x \in \mathbb{Z}_2^n} \zeta^{c_1 a(x)} \zeta^{c_2 b(x)} (-1)^{u \cdot x}. \tag{3.58}$$

Since $\zeta = e^{\frac{2\pi i}{q}}$, denoting $X_i = \cos \frac{2\pi c_i}{q}$ and $Y_i = \sin \frac{2\pi c_i}{q}$, $i = 1, 2$, we have

$$\zeta^{c_1 a(x)} = e^{\frac{2\pi i}{q} c_1 a(x)} = \left(\cos \frac{2\pi c_1}{q} + i \sin \frac{2\pi c_1}{q} \right)^{a(x)} = (X_1 + iY_1)^{a(x)},$$
$$\zeta^{c_2 b(x)} = e^{\frac{2\pi i}{q} c_2 b(x)} = \left(\cos \frac{2\pi c_2}{q} + i \sin \frac{2\pi c_2}{q} \right)^{b(x)} = (X_2 + iY_2)^{b(x)}.$$

Regarding the possible values of X_i and Y_i, $i = 1, 2$, we consider the following cases.

Case I: Let $X_i = 0$ and $Y_i = \pm 1$, $i = 1, 2$. The condition $X_i = 0$ is equivalent to $\cos \frac{2\pi c_i}{q} = 0$, i.e. $\frac{2\pi c_i}{q} = \frac{\pi}{2} + k_i \pi = \frac{\pi}{2}(1 + 2k_i)$, $k_i \in \mathbb{Z}$, $i = 1, 2$. Thus, we have $c_i = \frac{q}{4}(1 + 2k_i)$, $k_i \in \mathbb{Z}$, $i = 1, 2$, and since c_i, $i = 1, 2$, are integers (both from $\mathbb{Z}_q$), it must be the case that $q = 4s$, $s \in \mathbb{Z}^+$. We have

$$\zeta^{c_1 a(x)} \zeta^{c_2 b(x)} = \begin{cases} (-1)^{b(x)} i^{a(x)+b(x)}, & (Y_1, Y_2) = (1, -1) \\ (-1)^{a(x)} i^{a(x)+b(x)}, & (Y_1, Y_2) = (-1, 1) \\ (-1)^{a(x)+b(x)} i^{a(x)+b(x)}, & (Y_1, Y_2) = (-1, -1) \\ i^{a(x)+b(x)}, & (Y_1, Y_2) = (1, 1) \end{cases}$$

In order to handle (3.58) effectively, we need a suitable decomposition of the power of imaginary unit. In this section, we use $i^t = \frac{1+(-1)^t}{2} + \frac{1-(-1)^t}{2} i$, which holds if and only if t takes values $4l_1$ or $1 + 4l_2$, $l_i \in \mathbb{Z}$, $i = 1, 2$. Since $a(x) + b(x)$ is evaluated modulo $q = 4s$, and thus $a(x) + b(x) \in \{0, 1, 2\}$, we can not use this decomposition for further calculation of (3.58). The decomposition of i^t is of a great importance, since we want to express $\mathcal{H}_f(u)$ in terms of WHT coefficients of the functions a, b and $a+b$.

Case II: Let $X_i = \pm 1$ and $Y_i = 0$, $i = 1, 2$. Then $\frac{2\pi c_i}{q} = k_i \pi$, i.e. $c_i = \frac{q}{2} k_i$, $k_i \in \mathbb{Z}$, $i = 1, 2$. Since c_i are integers, q must be even and the function f is given by

$$f(x) = \begin{cases} \frac{q}{2} a(x), & (X_1, X_2) = (-1, 1) \\ \frac{q}{2} b(x), & (X_1, X_2) = (1, -1) \\ \frac{q}{2}(a(x) + b(x)), & (X_1, X_2) = (-1, -1) \\ 0, & (X_1, X_2) = (1, 1) \end{cases}$$

In the first three cases, it is not difficult to see that f is gbent if and only if a, b and $a + b$ are bent Boolean functions, respectively. The last case implies $c_i = 0$, $i = 1, 2$,

and f can not be a gbent function.

The following case appears to be the most interesting one, and the necessary conditions under which f is a gbent function are summarized in Proposition 3.5.1.

Case III: Let $X_1 = Y_2 = 0$ and $Y_1 = X_2 = \pm 1$. Then $X_1 = Y_2 = 0$ implies $c_1 = \frac{q}{4}(1 + 2k_1)$ and $c_2 = \frac{q}{2}k_2$, $k_i \in \mathbb{Z}$, $i = 1, 2$. Consequently, it must be the case that $q = 4s$, $s \in \mathbb{Z}^+$, with the following subcases:

(a) If $(Y_1, X_2) = (1, -1)$ or $(Y_1, X_2) = (-1, -1)$, then f is gbent iff b and $a + b$ are bent Boolean functions, where $c_2 = \frac{q}{2}$ (in both cases) and $c_1 = \frac{q}{4}$ or $c_1 = \frac{3q}{4}$, respectively (Proposition 3.5.1).

(b) If $(Y_1, X_2) = (-1, 1)$ or $(Y_1, X_2) = (1, 1)$, then using $\sum_{x \in \mathbb{Z}_2^n} (-1)^{u \cdot x} = \begin{cases} 2^n, & u = 0 \\ 0 & u \neq 0, \end{cases}$ it is not difficult to see that f can not be a gbent function.

Similarly, if $X_1 = Y_2 = \pm 1$ and $Y_1 = X_2 = 0$, we conclude that $q = 4s$, $s \in \mathbb{Z}^+$, with the following subcases:

(1) If $(X_1, Y_2) = (1, -1)$ or $(X_1, Y_2) = (1, 1)$, the scenario from the **Case III**-(b) is repeated.

(2) If $(X_1, Y_2) = (-1, 1)$ or $(X_1, Y_2) = (-1, -1)$, then f is gbent iff a and $a + b$ are bent Boolean functions, where $c_1 = \frac{q}{2}$ (in both cases) and $c_2 = \frac{q}{4}$ or $c_2 = \frac{3q}{4}$, respectively (Proposition 3.5.1).

The question whether there exist gbent functions when $\zeta^{c_1 a(x) + c_2 b(x)} \notin \{\pm 1, \pm i\}$ remains unanswered. Note that, the function f in the **Case II** takes only two values (namely 0 and $q/2$), and still is a gbent function from $\mathbb{Z}_2^n$ to $\mathbb{Z}_q$, where q is even.

3.5.1 Generalized bent functions from two bent Boolean functions

The following proposition deals with a gbent function represented as a linear combination of two Boolean functions in the form $f(x) = \frac{q}{2}a(x) + kb(x)$, where $k \in \{\frac{q}{4}, \frac{3q}{4}\}$. In addition, we have seen that in this case q must be equal to $4s$, $s \in \mathbb{Z}^+$.

Proposition 3.5.1 *Let $f \in \mathcal{GB}_q^n$, n even, $q = 4s$, $s \in \mathbb{Z}^+$, and $f(x) = \frac{q}{2}a(x) + kb(x)$, where $k \in \{\frac{q}{4}, \frac{3q}{4}\}$, $a, b \in \mathcal{B}_n$. Then, f is gbent if and only if a and $a + b$ are bent Boolean functions.*

PROOF: Let $f(x) = \frac{q}{2}a(x) + kb(x)$, where $k \in \{\frac{q}{4}, \frac{3q}{4}\}$, $a, b : \mathbb{Z}_2^n \to \mathbb{Z}_2$. Firstly, we consider the relation between the GWHT of f and the Walsh coefficients of a and $a + b$. This relation is derived in the same way as in [109, Lemma 31]. Since we are going to use some equalities later, for self-completeness we provide the details of the proof:

$$\mathcal{H}_f(\omega) = 2^{-\frac{n}{2}} \sum_{x \in \mathbb{Z}_2^n} \zeta^{f(x)} (-1)^{\omega \cdot x} = 2^{-\frac{n}{2}} \sum_{x \in \mathbb{Z}_2^n} \zeta^{\frac{q}{2}a(x)} \zeta^{kb(x)} (-1)^{\omega \cdot x}. \quad (3.59)$$

Further manipulation of the terms $\zeta^{\frac{q}{2}a(x)}$ and $\zeta^{kb(x)}$ gives the following,

$$\begin{aligned}\zeta^{\frac{q}{2}a(x)} &= e^{\frac{2\pi i}{q}\cdot\frac{q}{2}a(x)} = \left(e^{\pi i}\right)^{a(x)} = (\cos\pi + i\sin\pi)^{a(x)} = (-1)^{a(x)},\\ \zeta^{kb(x)} &= \begin{cases} i^{b(x)}, & k = \frac{q}{4}\\ (-i)^{b(x)}, & k = \frac{3q}{4}.\end{cases}\end{aligned}$$

In addition, since $b(x) \in \{0,1\}$, for every $x \in \mathbb{Z}_2^n$, we have $i^{b(x)} = \frac{1+(-1)^{b(x)}}{2} + \frac{1-(-1)^{b(x)}}{2}\cdot i$. We consider the calculation of (3.59) only for $k = \frac{q}{4}$, since due to the symmetry the same approach applies to the case $k = \frac{3q}{4}$:

$$\begin{aligned}\mathcal{H}_f(\omega) &= 2^{-\frac{n}{2}}\sum_{x\in\mathbb{Z}_2^n}(-1)^{a(x)}i^{b(x)}(-1)^{\omega\cdot x} =\\ &= 2^{-\frac{n}{2}}\sum_{x\in\mathbb{Z}_2^n}\left(\frac{1+(-1)^{b(x)}}{2} + \frac{1-(-1)^{b(x)}}{2}\cdot i\right)(-1)^{a(x)+\omega\cdot x} =\\ &= 2^{-\frac{n}{2}}\cdot\frac{1}{2}\left[\left(2^{\frac{n}{2}}W_a(\omega) + 2^{\frac{n}{2}}W_{a+b}(\omega)\right) + i\left(2^{\frac{n}{2}}W_a(\omega) - 2^{\frac{n}{2}}W_{a+b}(\omega)\right)\right] =\\ &= \frac{1}{2}\left[(W_a(\omega) + W_{a+b}(\omega)) + i\,(W_a(\omega) - W_{a+b}(\omega))\right]. \qquad (3.60)\end{aligned}$$

From (3.60), it follows that the relation between the Walsh-Hadamard transforms of functions f, a and $a+b$, which holds for every $\omega \in \mathbb{Z}_2^n$, satisfies the following,

$$|H_f(\omega)|^2 = \frac{1}{2}(W_a^2(\omega) + W_{a+b}^2(\omega)). \qquad (3.61)$$

If we assume that a and $a+b$ are bent Boolean functions, i.e., $|W_a(\omega)| = |W_{a+b}(\omega)| = 1$, then we have

$$|H_f(\omega)|^2 = \frac{1}{2}(W_a^2(\omega) + W_{a+b}^2(\omega)) = \frac{1}{2}(1+1) = 1.$$

This implies $|H_f(\omega)| = 1$, thus f is a gbent function.

On contrary, let us assume that f is generalized bent, i.e., $|H_f(\omega)| = 1$. Since the Walsh coefficients of an arbitrary Boolean function are integers and $W_a^2(\omega), W_{a+b}^2(\omega) \geq 0$, the equation (3.61) has a unique solution $W_a^2(\omega) = W_{a+b}^2(\omega) = 1$. It implies $W_a(\omega) = \pm 1$ and $W_{a+b}(\omega) = \pm 1$ for all $\omega \in \mathbb{Z}_2^n$, i.e., a and $a+b$ are bent Boolean functions. ∎

Note that, if we have a gbent function, say $g : \mathbb{Z}_2^n \to \mathbb{Z}_4$, $g(x) = b(x) + 2a(x)$, then we can obtain a gbent function $f = 2g$, $f : \mathbb{Z}_2^n \to \mathbb{Z}_8$, i.e. $f(x) = 2b(x) + 4a(x)$. Since g is a gbent function, by Proposition 3.5.1, it is equivalent to the fact that a and $a+b$ are bent Boolean functions. Furthermore, it means that f is a gbent function, since $f(x) = \frac{q}{2}a(x) + \frac{q}{4}b(x)$, for $q = 8$. Hence, with a function g we can obtain a gbent function $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, $q = 4s$, $s \in \mathbb{Z}^+$, just multiplying with the number s, i.e. $f(x) = sg(x)$.

3.5.2 Generalized bent functions using direct sum of gbent functions

We now consider a gbent function as a linear combination of gbent functions, with arbitrary coefficients. This generalizes the result derived in [108, Theorem 2] for two gbent functions, with a similar proof.

Proposition 3.5.2 *Suppose* $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, $q \geq 2$, $\mathbb{Z}_2^n = \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \times \cdots \times \mathbb{Z}_2^{n_r}$ *and the function* f *is given by*

$$f(x) = c_1 f_1(x^{(1)}) + c_2 f_2(x^{(2)}) + \cdots + c_r f_2(x^{(r)}),$$

where $x = (x^{(1)}, x^{(2)}, \ldots, x^{(r)}) \in \mathbb{Z}_2^n$, $x^{(i)} \in \mathbb{Z}_2^{n_i}$, $f_i : \mathbb{Z}_2^{n_i} \to \mathbb{Z}_p$, $2 \leq p \leq q$, $c_i \in \mathbb{Z}_q$, $i = 1, 2, \ldots, r$. *Then,* f *is a gbent function if any only if* f_i *are gbent functions, for every* $i = 1, 2, \ldots, r$.

Proof: Let $u \in \mathbb{Z}_2^n = \mathbb{Z}_2^{n_1} \times \mathbb{Z}_2^{n_2} \times \cdots \times \mathbb{Z}_2^{n_r}$, $u = (u_1, u_2, \ldots, u_r)$, $u_i \in \mathbb{Z}_2^{n_i}$, $i = 1, 2, \ldots, r$, be arbitrary. For any $u \in \mathbb{Z}_2^n$, the Walsh-Hadamard coefficient at u is given by

$$\begin{aligned} 2^n \mathcal{H}_f(u) &= \sum_{x \in \mathbb{Z}_2^n} \zeta^{f(x)} (-1)^{u \cdot x} = \sum_{x^{(1)} \in \mathbb{Z}_2^{n_1}} \zeta^{c_1 f_1(x^{(1)})} (-1)^{u_1 \cdot x^{(1)}} \cdots \sum_{x^{(r)} \in \mathbb{Z}_2^{n_r}} \zeta^{c_r f_r(x^{(r)})} (-1)^{u_r \cdot x^{(r)}} \\ &= \left(2^{\frac{n_1}{2}} \zeta^{c_1} \mathcal{H}_{f_1}(u_1)\right) \left(2^{\frac{n_2}{2}} \zeta^{c_2} \mathcal{H}_{f_2}(u_2)\right) \cdots \left(2^{\frac{n_r}{2}} \zeta^{c_r} \mathcal{H}_{f_r}(u_r)\right). \end{aligned}$$

Suppose that f_i are gbent functions, $i = 1, 2, \ldots, r$. Since $2^n = 2^{n_1+n_2+\cdots+n_r}$, $|\zeta^{c_1+c_2+\cdots+c_r}| = 1$ and the codomain of f_i does not affect the value $|\mathcal{H}_{f_i}(u_i)|$ (f_i are gbent), i.e., $|\mathcal{H}_{f_i}(u_i)| = 1$ for every $i = 1, 2, \ldots, r$. Thus,

$$|\mathcal{H}_f(u)| = |\zeta^{c_1+c_2+\cdots+c_r}| \prod_{i=1}^{r} |\mathcal{H}_{f_i}(u_i)| = 1, \tag{3.62}$$

for every $u \in \mathbb{Z}_2^n$, regardless of whether $|\mathcal{H}_{f_i}(u)| \in \mathbb{C}$ or $|\mathcal{H}_{f_i}(u)| \in \mathbb{R}$ (using the properties of complex numbers). It implies that f is a gbent function.

Conversely, let f be a gbent function. We need to prove that for every $i = 1, 2, \ldots, r$, f_i is a gbent function. From (3.62), we have

$$1 = |\mathcal{H}_f(u)| = \prod_{i=1}^{r} |\mathcal{H}_{f_i}(u_i)|.$$

Let us assume that there exists a function f_{i_0}, $1 \leq i_0 \leq r$, which is not a gbent function. It means that there exists an element $u_{i_0} \in \mathbb{Z}_2^{n_{i_0}}$ such that $|\mathcal{H}_{f_{i_0}}(u_{i_0})| = t \neq 1$ ($t \neq 0$). Then, if we for instance consider $\mathcal{H}_{f_1}(u_1)$, we have $|\mathcal{H}_{f_1}(u_1)| = \frac{1}{kt}$, where $k \geq 1$, and k is equal to the product of $|\mathcal{H}_{f_i}(u_i)|$, for $i = 1, 2, \ldots, r$ and $i \notin \{1, i_0\}$. Regardless the number k, we have

$$\sum_{u_1 \in \mathbb{Z}_2^{n_1}} |\mathcal{H}_{f_1}(u_1)|^2 = \frac{2^{n_1}}{(kt)^2} \neq 2^{n_1},$$

which is a contradiction, by the generalized Parseval's identity. It follows that f_i are gbent functions, for every $i = 1, 2, \ldots, r$. ∎

3.5.3 Generalized bent functions from one bent and one gbent function

Finally, we consider the third case when a gbent function $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, n even, $q = 4 \cdot s$, $s \in \mathbb{Z}^+$, is represented as a linear combination of one generalized $a \in \mathcal{GB}_q^n$ and one bent function $b \in \mathcal{B}_n$ of the form $f = \frac{q}{2}a + kb$, $k \in \{\frac{q}{4}, \frac{3q}{4}\}$.

Proposition 3.5.3 *Suppose that $f(x) = \frac{q}{2}a(x) + kb(x)$, $a \in \mathcal{GB}_q^n$, $b \in \mathcal{B}_n$, $k \in \{\frac{q}{4}, \frac{3q}{4}\}$ and let $h_1, h_2 \in \mathcal{B}_n$ be two arbitrary bent functions. If $(-1)^{a(x)} = (-1)^{h_1(x)}$ and $(-1)^{a(x)+b(x)} = (-1)^{h_2(x)}$, for every $x \in \mathbb{Z}_2^n$, then f is a gbent function.*

Proof: Note that the function $a + b$ can be considered as a generalized function, since $a(x) + b(x) \in \mathbb{Z}_q$. The conditions $(-1)^{a(x)} = (-1)^{h_1(x)}$ and $(-1)^{a(x)+b(x)} = (-1)^{h_2(x)}$, for every $x \in \mathbb{Z}_2^n$, mean that the even values of the function a and $a + b$ correspond to zero values of the functions h_1 and h_2 (odd values correspond to ones) over $\mathbb{Z}_2^n$. Using the proof of Proposition 3.5.1, $(-1)^{a(x)} = (-1)^{h_1(x)}$, $(-1)^{a(x)+b(x)} = (-1)^{h_2(x)}$, for any $k \in \{\frac{q}{4}, \frac{3q}{4}\}$ we have

$$\begin{aligned} |\mathcal{H}_f(u)\ |^2 &= \frac{1}{2}\left[\left(\sum_{x\in\mathbb{Z}_2^n}(-1)^{a(x)+u\cdot x}\right)^2 + \left(\sum_{x\in\mathbb{Z}_2^n}(-1)^{a(x)+b(x)+u\cdot x}\right)^2\right] \qquad (3.63)\\ &= \frac{1}{2}(H_{h_1}^2(u) + H_{h_2}^2(u)) = 1, \end{aligned}$$

which implies that f is a gbent function. ∎

Proposition 3.5.4 *Suppose that $f(x) = \frac{q}{2}a(x) + \frac{q}{4}b(x)$, $a \in \mathcal{B}_n$, $b \in \mathcal{GB}_q^n$, $q = 4s$, $s \in \mathbb{Z}^+$. If $b(x)$ takes values $4l_1$ or $1 + 4l_2$ when $x \in \mathbb{Z}_2^n$, $l_i \in \mathbb{Z}$, $i = 1, 2$, and*

$$(-1)^{a(x)+b(x)} = (-1)^{h(x)} \quad \text{for all } x \in \mathbb{Z}_2^n,$$

for an arbitrary bent function $h \in \mathcal{B}_n$, then f is a gbent function in $\mathcal{GB}_q^n$.

Proof: Since $\zeta^{\frac{q}{2}a(x)} = (-1)^{a(x)}$ and $\zeta^{\frac{q}{4}b(x)} = i^{b(x)}$, the equation $i^{b(x)} = \frac{1+(-1)^{b(x)}}{2} + \frac{1-(-1)^{b(x)}}{2} \cdot i$, is possible iff $b(x)$ takes values $4l_1$ or $1 + 4l_2$ over $\mathbb{Z}_2^n$, for some $l_i \in \mathbb{Z}$ which may depend on x. Consequently, the equation (3.63) holds and therefore $|\mathcal{H}_f(u)|^2 = \frac{1}{2}(H_a^2(u) + H_h^2(u)) = 1$, for every $u \in \mathbb{Z}_2^n$. ∎

Remark 3.5.5 *Notice that for $k = \frac{3q}{4}$, we would not be able to make a condition on the values of b, such that we can decompose $i^{b(x)}$ as above. In addition, notice that the values $4l_1$ of the function $b(x)$, evaluated modulo $q = 4s$, are not always equal to zero, since we may have $l_1 < s$ implying that $4l_1$ may take other values than 0.*

3.5.4 Concatenation of generalized bent Boolean functions

In what follows, we give some theoretical results regarding some secondary constructions of generalized bent functions. Using a concatenation of generalized bent functions, on even and odd numbers of variables, we give a more generalized approach of construction which essentially relies on Proposition 3.5.1. On contrary to recently proposed constructions [108, 111, 113] that specify the conditions on the WHT coefficients of constituent functions (also called restrictions) used in the concatenation for particular cases, we show that the constituent functions belong to a more general form, the one mentioned in the Proposition 3.5.1. For the purpose of providing some practical construction methods and easier overview of WHT conditions, we use only the Walsh-Hadamard formula, even though the same conditions may be turned into equivalent conditions expressed by means of cross-correlation.

3.5.5 Generalized bent functions defined on $\mathbb{Z}_2^n$, n odd

In this section we assume that $f : \mathbb{Z}_2^{n+1} \to \mathbb{Z}_q$, $q = 4s$, $s \in \mathbb{Z}^+$ and n even. The space $\mathbb{Z}_2^{n+1}$ is identified with $\mathbb{Z}_2^n \times \mathbb{Z}_2$. We start with a general result regarding the properties of the restrictions of gbent functions.

Proposition 3.5.6 *Let the restrictions of f on $\mathbb{Z}_2^n$ be given by,*

$$f(x,y) = \begin{cases} f_1(x) + C_1, & y = 0 \\ f_2(x) + C_2, & y = 1 \end{cases}$$

where $f_1, f_2 \in \mathcal{GB}_q^n$, $q = 4s$, $s \in \mathbb{Z}^+$, n even. If $C_1 - C_2 = s(1+2t)$, for some $t \in \mathbb{Z}$, then f is gbent if and only if f_1 and f_2 are gbent functions.

Proof: The GWHT of f is given by

$$2^{\frac{n+1}{2}} |\mathcal{H}_f(u,v)|^2 = 2^{\frac{n}{2}} \zeta^{C_1} H_{f_1}(u) + (-1)^v 2^{\frac{n}{2}} \zeta^{C_2} H_{f_2}(u),$$

for every $u \in \mathbb{Z}_2^n$, $v \in \mathbb{Z}_2$. The absolute value of $\mathcal{H}_f(u,v)$ is given by

$$2|\mathcal{H}_f(u,v)|^2 = H_{f_1}^2(u) + H_{f_2}^2(u) + 2(-1)^v H_{f_1}(u) H_{f_2}(u) \cos(\varphi - \psi),$$

where $\varphi = \frac{\pi C_1}{2s}$ and $\psi = \frac{\pi C_2}{2s}$. Since $C_1 - C_2 = s(1+2t)$, i.e. $\varphi - \psi = \frac{\pi}{2} + t\pi$, $t \in \mathbb{Z}$. It means that $\cos(\varphi - \psi) = 0$, and then $2|\mathcal{H}_f(u,v)|^2 = H_{f_1}^2(u) + H_{f_2}^2(u)$. This equation implies that f is gbent if and only if f_1 and f_2 are gbent, since $H_{f_i}(u) \in \mathbb{Z}$, for every $u \in \mathbb{Z}_2^n$. ∎

Remark 3.5.7 *Proposition 3.5.6 can be easily generalized for $f \in \mathcal{GB}_q^{n+m}$, where generalized constituent functions are taken from $\mathcal{GB}_q^n$, $q = 4s$, $s \in \mathbb{Z}$, n even. Regarding the condition on constants C_i, we may state the proposition for $q \geq 2$, q even, but the conditions would then change.*

A generalized Boolean function $f \in \mathcal{GB}_q^{n+2}$ is symmetric with respect to two variables y and z if and only if there exist $g, h, s \in \mathcal{GB}_q^n$ such that

$$f(x,y,z) = g(x) + (y \oplus z)h(x) + yzs(x), \tag{3.64}$$

where $x \in \mathbb{Z}_2^n$, $y, z \in \mathbb{Z}_2$, and $\mathbb{Z}_2^{n+2} = \mathbb{Z}_2^n \times \mathbb{Z}_2 \times \mathbb{Z}_2$. The binary case was investigated in [122] and a generalization of their main result was later addressed in [111, 112]. On the other hand, Proposition 3.5.6 and Remark 3.5.7 generalize such constructions on $\mathcal{GB}_q^{n+m}$, n even, where q depends on the choice of the constants.

In what follows, we show that if the restriction of f to $\mathbb{Z}_2^n$, with respect to the values $y \in \mathbb{Z}_2$, are of the form $\frac{q}{2}A(x)+kB(x)+C$, for some Boolean functions $A, B \in \mathcal{B}_n$, $C \in \mathbb{Z}_q$, then we may obtain a gbent function by setting certain conditions on the GWHT coefficients of the constituent functions A and B, or/and on the constant C. Since there are two possible values for the parameter $y \in \mathbb{Z}_2$, and therefore two possible restrictions for the function f, we denote $a_1 = W_{A_1}(u)$, $b_1 = W_{A_1+B_1}(u)$, $a_2 = W_{A_2}(u)$, $b_2 = W_{A_2+B_2}(u)$, for $u \in \mathbb{Z}_2^n$, and $\varphi = \frac{\pi C_1}{2s}$, $\psi = \frac{\pi C_2}{2s}$. Since most of the calculations are straightforward computation of the WHT, we only provide the full proof of Theorem 3.5.9.

Remark 3.5.8 *In the following two theorems, we assume that $a_i^2(u) = b_i^2(u) = 1$, for all $u \in \mathbb{Z}_2^n$, i.e., the functions A_i, B_i and $A_i + B_i$ are bent in $\mathcal{B}_n$, $i = 1, 2$.*

Theorem 3.5.9 *Let the restrictions of f on $\mathbb{Z}_2^n$ be given by*

$$f(x,y) = \begin{cases} \frac{q}{2}A_1(x) + kB_1(x) + C_1, & y = 0, \\ \frac{q}{2}A_2(x) + kB_2(x) + C_2, & y = 1, \end{cases} \tag{3.65}$$

where $k = \frac{q}{4}$ or $\frac{3q}{4}$, and assume the equality

$$(a_1a_2 + b_1b_2)\cos(\varphi - \psi) + (-1)^{\frac{4k}{q}}(a_2b_1 - a_1b_2)\sin(\varphi - \psi) = 0, \tag{3.66}$$

holds. Then f is a gbent function.

Proof: Using (3.60) from Proposition 3.5.1 and choosing $k = \frac{q}{4}$, we compute the Walsh-Hadamard coefficients at $(u, v) \in \mathbb{Z}_2^n \times \mathbb{Z}_2$:

$$\begin{aligned} 2^{\frac{n+1}{2}}\mathcal{H}_f(u,v) &= \sum_{(x,y)\in\mathbb{Z}_2^{n+1}} \zeta^{f(x,y)}(-1)^{u\cdot x\oplus vy} = \sum_{x\in\mathbb{Z}_2^n}\sum_{y\in\mathbb{Z}_2} \zeta^{f(x,y)}(-1)^{u\cdot x\oplus vy} \\ &= \sum_{x\in\mathbb{Z}_2^n} \left(\zeta^{f(x,0)}(-1)^{u\cdot x} + \zeta^{f(x,1)}(-1)^{u\cdot x\oplus v}\right) \\ &= \sum_{x\in\mathbb{Z}_2^n} \zeta^{f(x,0)}(-1)^{u\cdot x} + \sum_{x\in\mathbb{Z}_2^n} \zeta^{f(x,1)}(-1)^{u\cdot x\oplus v} \\ &= \frac{1}{2}2^{\frac{n}{2}}\zeta^{C_1}\left[(W_{A_1}(u) + W_{A_1+B_1}(u)) + i(W_{A_1}(u) - W_{A_1+B_1}(u))\right] \\ &\quad + \quad (-1)^v\frac{1}{2}2^{\frac{n}{2}}\zeta^{C_2}\left[(W_{A_2}(u) + W_{A_2+B_2}(u)) + i(W_{A_2}(u) - W_{A_2+B_2}(u))\right]. \end{aligned}$$

Using $a_1 = W_{A_1}(u)$, $b_1 = W_{A_1+B_1}(u)$, $a_2 = W_{A_2}(u)$ and $b_2 = W_{A_2+B_2}(u)$, we get

$$2\sqrt{2}\mathcal{H}_f(u,v) = \zeta^{C_1}\left[(a_1 + b_1) + i(a_1 - b_1)\right] + (-1)^v\zeta^{C_2}\left[(a_2 + b_2) + i(a_2 - b_2)\right]. \tag{3.67}$$

Then, for $k = \frac{3q}{4}$ we have

$$2\sqrt{2}\mathcal{H}_f(u,v) = \zeta^{C_1}\left[(b_1+a_1)+i(b_1-a_1)\right] + (-1)^v\zeta^{C_2}\left[(b_2+a_2)+i(b_2-a_2)\right]. \tag{3.68}$$

Since $\zeta^{C_1} = e^{\frac{2\pi i}{4s}C_1} = \cos\frac{\pi C_1}{2s} + i\sin\frac{\pi C_1}{2s} = \cos\varphi + i\sin\varphi$, where $\varphi = \frac{\pi C_1}{2s}$ and $\zeta^{C_2} = e^{\frac{2\pi i}{4s}C_2} = \cos\frac{\pi C_2}{2s} + i\sin\frac{\pi C_2}{2s} = \cos\psi + i\sin\psi$, where $\psi = \frac{\pi C_2}{2s}$, the equation (3.67) is equivalent to

$$\begin{aligned}2\sqrt{2}\mathcal{H}_f(u,v) = {} & \left[((a_1+b_1)\cos\varphi - (a_1-b_1)\sin\varphi) + (-1)^v((a_2+b_2)\cos\psi - (a_2-b_2)\sin\psi)\right] \\ & + i\left[((a_1-b_1)\cos\varphi + (a_1+b_1)\sin\varphi) + (-1)^v((a_2-b_2)\cos\psi + (a_2+b_2)\sin\psi)\right].\end{aligned} \tag{3.69}$$

From (3.69), we know $\mathfrak{Re}(\mathcal{H}_f(u,v))$ and $\mathfrak{Im}(\mathcal{H}_f(u,v))$. Now, using (3.69) we have

$$2\sqrt{2}|\mathcal{H}_f(u,v)| = \sqrt{(\mathfrak{Re}(\mathcal{H}_f(u,v)))^2 + (\mathfrak{Im}(\mathcal{H}_f(u,v)))^2}, \quad i.e.$$

$$\begin{aligned}4|\mathcal{H}_f(u,v)|^2 = {} & (a_1^2+a_2^2+b_1^2+b_2^2) + 2(-1)^v\left[(a_1a_2+b_1b_2)\cos(\varphi-\psi)\right. \\ & \left.+(a_2b_1-a_1b_2)\sin(\varphi-\psi)\right].\end{aligned} \tag{3.70}$$

For $k = \frac{3q}{4}$, we have

$$\begin{aligned}4|\mathcal{H}_f(u,v)|^2 = {} & (a_1^2+a_2^2+b_1^2+b_2^2) + 2(-1)^v\left[(a_1a_2+b_1b_2)\cos(\varphi-\psi)\right. \\ & \left.-(a_2b_1-a_1b_2)\sin(\varphi-\psi)\right].\end{aligned} \tag{3.71}$$

Since A_i, B_i, A_i+B_i are bent Boolean functions (Remark 3.5.8), $i = 1,2$, we have $a_1^2 = a_2^2 = b_1^2 = b_2^2 = 1$. Moreover (3.66) holds, and by (3.70) and (3.71) we have $4|\mathcal{H}_f(u,v)|^2 = a_1^2+a_2^2+b_1^2+b_2^2 = 4$, i.e., $|\mathcal{H}_f(u,v)|^2 = 1$ which implies that f is a gbent function. ∎

Remark 3.5.10 *Regarding the equation (3.66) in Theorem 3.5.9, we consider the following cases:*

Case 1: *If $a_2b_1 \neq a_1b_2$ for every $u \in \mathbb{Z}_2^n$, φ and ψ is such that $\cos(\varphi-\psi) \neq 0$, and $a_i^2 = b_i^2 = 1$, $i = 1,2$, then the possible values for a_i and b_i which satisfy $a_2b_1 \neq a_1b_2$ imply that $\varphi - \psi = t\pi$, where $t \in \mathbb{Z}$. Note that $\varphi - \psi = t\pi$ satisfies the condition $\cos(\varphi-\psi) \neq 0$. Since $\varphi = \frac{\pi C_1}{2s}$ and $\psi = \frac{\pi C_2}{2s}$, we have $C_1 - C_2 = 2st$, $t \in \mathbb{Z}$.*

Case 2: *If $a_2b_1 = a_1b_2$ holds, then the equation (3.66) is equivalent to $(a_1a_2 + b_1b_2)\cos(\varphi-\psi) = 0$. This equation holds if $a_1a_2 = -b_1b_2$ or $\cos(\varphi-\psi) = 0$. The first condition is related to the Walsh-Hadamard coefficients of the functions A_i, B_i and A_i+B_i, $i = 1,2$. The second condition implies $\varphi - \psi = \frac{\pi}{2} + t\pi$, i.e. $C_1 - C_2 = s(1+2t)$, $t \in \mathbb{Z}$.*

Hence, regarding the restrictions of the function f given by (3.65) and Remark 3.5.10, we have a set of conditions which can be imposed on the constituent functions and constants so that f is a gbent function.

Theorem 3.5.11 *Let the restrictions of f on $\mathbb{Z}_2^n$ be given as*

$$f(x,y) = \begin{cases} \frac{q}{2}A_1(x) + kB_1(x) + C_1, & y = 0 \\ \frac{q}{2}A_2(x) + C_2, & y = 1, \end{cases}$$

where $k = \frac{q}{4}$ or $\frac{3q}{4}$, and assume that the equality

$$(a_1 + b_1)\cos(\varphi - \psi) + (-1)^{\frac{4k}{q}-1}(a_1 - b_1)\sin(\varphi - \psi) = 0 \tag{3.72}$$

holds for any $u \in \mathbb{Z}_2^n$. Then, the function f is a gbent function.

Remark 3.5.12 *If the restrictions of f on $\mathbb{Z}_2^n$ are given as*

$$f(x,y) = \begin{cases} \frac{q}{2}A_1(x) + C_1, & y = 0 \\ \frac{q}{2}A_2(x) + kB_2(x) + C_2, & y = 1, \end{cases}$$

then the equation (3.72) is replaced with

$$(a_2 + b_2)\cos(\varphi - \psi) + (-1)^{\frac{4k}{q}}(a_2 - b_2)\sin(\varphi - \psi) = 0, \quad \forall u \in \mathbb{Z}_2^n. \tag{3.73}$$

Theorem 3.5.13 *Let the restrictions of f on $\mathbb{Z}_2^n$ be given by,*

$$f(x,y) = \begin{cases} \frac{q}{2}A_1(x) + C_1, & y = 0 \\ \frac{q}{2}A_2(x) + C_2, & y = 1, \end{cases}$$

and for some $t \in \mathbb{Z}$ let $C_1 - C_2 = s(1 + 2t)$. Then, f is a gbent function if and only if A_1 and A_2 are bent Boolean functions.

3.5.6 Generalized bent functions defined on $\mathbb{Z}_2^n$, n even

We now consider a function $f : \mathbb{Z}_2^{n+2} \to \mathbb{Z}_q$, $q = 4s$, $s \in \mathbb{Z}^+$, n even. Since the space $\mathbb{Z}_2^{n+2}$ is identified with $\mathbb{Z}_2^n \times \mathbb{Z}_2 \times \mathbb{Z}_2$, there will be two parameters, say y and z, i.e., $f = f(x,y,z)$, $x \in \mathbb{Z}_2^n$, $(y,z) \in \mathbb{Z}_2^2$. Since there are two possible forms for the restrictions of f, namely $\frac{q}{2}A(x) + kB(x) + C$ and $\frac{q}{2}A(x) + C$, and four different restrictions of the function f depending on the choice of $(y,z) \in \mathbb{Z}_2^2$, thus in total we will have at least 2^4 ways to obtain a gbent function, by setting certain conditions on the Walsh-Hadamard coefficients of the restrictions.

Hence, to obtain a gbent function with certain conditions, we need to derive the GWHT relation between the function f and its constituent functions. In the previous section, we have seen that these relations may give more solutions, involving the WHT equalities of the constituent functions and their constants. In this context, imposing only the conditions on constants implies the following.

Theorem 3.5.14 *Let the restrictions of f on $\mathbb{Z}_2^n$ be given by $\frac{q}{2}A_i(x) + C_i$, with respect to $(y,z) \in \mathbb{Z}_2^2$, $i = 1, \ldots, 4$, and all equations $C_i - C_j = s(1 + 2t_r)$ hold, for every $1 \le i < j \le 4$, where $t_r \in \mathbb{Z}$, $r = 1, \ldots, 4$. Then, f is a gbent function if and only if $A_i \in \mathcal{B}_n$ are bent, $i = 1, \ldots, 4$.*

PROOF: The GWHT of f is given by

$$2^{\frac{n+2}{2}}\mathcal{H}_f(u,v,w) = \sum_{(x,y,z)\in\mathbb{Z}_2^{n+2}} \zeta^{f(x,y,z)}(-1)^{u\cdot x\oplus vy\oplus wz} = \sum_{x\in\mathbb{Z}_2^n}\sum_{(y,z)\in\mathbb{Z}_2^2} \zeta^{f(x,y,z)}(-1)^{u\cdot x\oplus vy\oplus wz} =$$
$$= 2^{\frac{n}{2}}\left[\zeta^{C_1}W_{A_1}(u) + (-1)^w\zeta^{C_2}W_{A_2}(u) + (-1)^v\zeta^{C_3}W_{A_3}(u) + (-1)^{v\oplus w}\zeta^{C_4}W_{A_4}(u)\right].$$

Denoting by $a_i = W_{A_i}(u)$, $\varphi_i = \frac{\pi C_i}{2s}$, $i = 1,2,3,4$, where $\zeta^{C_i} = \cos\frac{\pi C_i}{2s} + i\sin\frac{\pi C_i}{2s}$, we have

$$\begin{aligned}4|\mathcal{H}_f(u,v,w)|^2 =& (a_1^2+a_2^2+a_3^2+a_4^2) + 2(-1)^w\left[a_1a_2\cos(\varphi_1-\varphi_2) + a_3a_4\cos(\varphi_3-\varphi_4)\right] + \\ & 2(-1)^v\left[a_1a_3\cos(\varphi_1-\varphi_3) + a_2a_4\cos(\varphi_2-\varphi_4)\right] + \\ & 2(-1)^{v\oplus w}\left[a_2a_3\cos(\varphi_2-\varphi_3) + a_1a_4\cos(\varphi_1-\varphi_4)\right].\end{aligned}$$

Since the equations $C_i - C_j = s(1+2t_r)$ hold, we have $\cos(\varphi_i-\varphi_j) = 0$, for every $1 \le i < j \le 4$, and $t_r \in \mathbb{Z}$, $r = 1,\ldots,4$. Therefore, the equation

$$\begin{aligned}&2(-1)^w\left[a_1a_2\cos(\varphi_1-\varphi_2) + a_3a_4\cos(\varphi_3-\varphi_4)\right] + 2(-1)^v\left[a_1a_3\cos(\varphi_1-\varphi_3) + \right. \\ &\left. a_2a_4\cos(\varphi_2-\varphi_4)\right] + 2(-1)^{v\oplus w}\left[a_2a_3\cos(\varphi_2-\varphi_3) + a_1a_4\cos(\varphi_1-\varphi_4)\right] = 0\end{aligned}$$

holds. It implies $4|\mathcal{H}_f(u,v,w)|^2 = a_1^2+a_2^2+a_3^2+a_4^2$, and since $a_i \in \mathbb{Z}$, we see that f is gbent iff $A_i \in \mathcal{B}_n$ are bent. ∎

Remark 3.5.15 *Theorem 3.5.13 and Theorem 3.5.14 can be easily generalized for $f \in \mathcal{GB}_q^{n+m}$, defined on the space $\mathbb{Z}_2^{n+m}$, n even. In such a situation, the system of equations concerning the constants C_i, $i = 1,2,\ldots,m$, will consist of $\binom{m}{2}$ equations in variables $C_i \in \mathbb{Z}_q$.*

3.5.7 Construction methods for generalized bent functions in $\mathcal{GB}_n^{4s}$

In this section we describe two constructions using theorems given in Section 3.5. In addition, we show that some constructions proposed in [108, 107, 111, 113] are just special cases of the results given in the previous section, with some particular conditions on its component functions or constants.

The next theorem is given in [111], and it deals with functions from $\mathbb{Z}_2^{n+1}$ to $\mathbb{Z}_8$.

Theorem 3.5.16 *[111] Let $f : \mathbb{Z}_2^{n+1} \to \mathbb{Z}_8$ (n is even) be given by*

$$f(x,y) = 4c(x) + (4a(x) + 4c(x) + 2\epsilon)y,$$

where $\epsilon \in \{-1,1\}$. Then, f is gbent in $\mathcal{GB}_8^{n+1}$ if and only if a, c are bent in $\mathcal{B}_n$. Moreover, if g is given by

$$g(x,y) = 4c(x) + (4a(x) + 2c(x) + 2\epsilon)y,$$

where $\epsilon \in \{-1,1\}$, $a, c \in \mathcal{B}_n$ such that $a, c, a+c$ are all bent, then g is gbent in $\mathcal{GB}_8^{n+1}$. Further, let h be given by

$$h(x,y) = 4c(x) + (4a(x) + 2\epsilon)y,$$

where $\epsilon \in \{-1,1\}$. Then, h is gbent in $\mathcal{GB}_8^{n+1}$ if and only if $c, a+c$ are bent in $\mathcal{B}_n$.

It is not difficult to see that the functions f and h above are related to Theorem 3.5.13, whereas g relates to the decomposition in Remark 3.5.12. For instance, let us consider the function f, whose restrictions are given by

$$f(x,y) = \begin{cases} 4c(x) = \frac{q}{2}c(x), & y = 0 \\ 4a(x) + 2\epsilon = \frac{q}{2}a(x) + 2\epsilon, & y = 1, \end{cases}$$

implying that $A_1(x) = c(x)$, $A_2(x) = a(x)$, $C_1 = 0$, $C_2 = 2\epsilon$, $\epsilon \in \{-1, 1\}$, in Theorem 3.5.11. If $\epsilon = -1$, then for $t = 0$ we have that $C_1 - C_2 = -2\epsilon = s(1 + 2t)$ holds, since from $q = 8 = 4s$ we have $s = 2$. For $\epsilon = 1$ and $t = -1$, $C_1 - C_2 = 2\epsilon = s(1 + 2t)$ holds, and therefore we have that f is gbent iff $A_1(x) = c(x)$ and $A_2(x) = a(x)$ are bent Boolean functions.

When the function h is of concern, we have that $A_1(x) = c(x)$ and $A_2(x) = a(x) + c(x)$, and the statements of Theorem 3.5.16 hold for the same values of $t \in \mathbb{Z}$.

The restrictions of the function g are given by,

$$g(x,y) = \begin{cases} 4c(x) = \frac{q}{2}c(x), & y = 0 \\ 4(a(x) + c(x)) + 2c(x) + 2\epsilon = \frac{q}{2}(a(x) + c(x)) + \frac{q}{4}c(x) + 2\epsilon, & y = 1, \end{cases}$$

so that $A_1(x) = B_2(x) = c(x)$, $A_2(x) = a(x) + c(x)$, $C_1 = 0$, $C_2 = 2\epsilon$, in Remark 3.5.12 and the condition (3.73) is used. Since $s = 2$, we have $\varphi = \frac{\pi C_1}{2s} = 0$ and $\psi = \frac{\pi C_2}{2s} = \pm\frac{\pi}{2}$, where the choice of "$\pm$" depends on the value of the parameter ϵ. Then, the equation (3.73) is equivalent to $a_2 - b_2 = 0$, since the GWHT of the function g is given by

$$4|H_g(u,v)|^2 = (2a_1^2 + a_2^2 + b_2^2) - 2\epsilon(-1)^v a_1(a_2 - b_2).$$

We have that g is gbent iff $a_1^2 = a_2^2 = b_2^2 = 1$ and $a_2 = b_2$, i.e. c, $a + 2c = a$ and $a + c$ are bent Boolean functions and $W_{a+c}(u) = W_a(u)$ holds, for every $u \in \mathbb{Z}_2^n$.

Remark 3.5.17 *Regarding the condition $W_{a+c}(u) = W_a(u)$ above, it seems to be omitted for the function g in [111, 113, Theorem 26].*

Providing that the conditions in theorems given in Section 3.5 are satisfied, we are able to obtain many gbent functions, with particular conditions on its constituent functions. For instance, we can give the following construction, which uses Theorem 3.5.9.

Example 3.5.18 *Let $f : \mathbb{Z}_2^{n+1} \to \mathbb{Z}_q$, $q = 4s$ and n even. Let restrictions of the function f be*

$$f(x,y) = \begin{cases} \frac{q}{2}(a(x) + b(x)) + \frac{q}{4}c(x) + C_1, & y = 0 \\ \frac{q}{2}b(x) + \frac{q}{4}(b(x) + c(x)) + C_2, & y = 1 \end{cases} \tag{3.74}$$

where $A_1(x) = a(x) + b(x)$, $A_2(x) = b(x)$, $B_1(x) = c(x)$, $B_2(x) = b(x) + c(x)$, $C_1 - C_2 = 2st$, $t \in \mathbb{Z}$ arbitrary. If we assume that $b, c, a + b, b + c$ are bent Boolean functions and $W_b(u)W_{a+b}(u) + W_{a+b+c}(u)W_c(u) = 0$, for every $u \in \mathbb{Z}_2^{n+1}$, then f is a gbent function.

To derive the explicit form, we assume that the function f is given in the form

$$f(x,y)=\alpha(y)a(x)+\beta(y)b(x)+\gamma(y)c(x)+\delta(y),$$

where we need to determine α,β,γ and δ from (3.74). Since the functions α,β,γ and δ in (3.74) are linear functions in y, their forms are easy to obtain. We get $\alpha(y)=-\frac{q}{2}y+\frac{q}{2}$, $\beta(y)=\frac{q}{4}y+\frac{q}{2}$, $\gamma(y)=\frac{q}{4}$ and $\delta(y)=(C_2-C_1)y+C_1$, $t\in\mathbb{Z}$. The function f is given by

$$f(x,y)=(-\frac{q}{2}y+\frac{q}{2})a(x)+(\frac{q}{4}y+\frac{q}{2})b(x)+\frac{q}{4}c(x)+(C_2-C_1)y+C_1,$$

where $q=4s$, $s\in\mathbb{Z}^+$, $C_1-C_2=2st$, $t\in\mathbb{Z}$.

The following example provides a construction of gbent functions defined on $\mathbb{Z}_2^{n+2}$, n even. Notice that there are many other gbent functions that can be derived similarly using different coefficients and constants that satisfy gbent conditions mentioned earlier.

Example 3.5.19 *Let $f:\mathbb{Z}_2^{n+2}\to\mathbb{Z}_q$, $q=4s$ and n even. Let the restrictions of the function f be given by*

$$f(x,y,z)=\begin{cases}\frac{q}{2}a(x)+C_1, & (y,z)=(0,0)\\ \frac{q}{2}b(x)+\frac{q}{4}(a(x)+b(x))+C_2, & (y,z)=(0,1)\\ \frac{q}{2}a(x)+\frac{3q}{4}c(x)+C_3, & (y,z)=(1,0)\\ \frac{q}{2}c(x)+C_4, & (y,z)=(1,1),\end{cases}\tag{3.75}$$

where $A_1(x)=A_3(x)=a(x)$, $A_2(x)=b(x)$, $B_2(x)=a(x)+b(x)$, $B_3(x)=A_4(x)=c(x)$. Also, $C_i-C_j=s(1+2t_r)$, $1\le i<j\le 4$ and $t_r\in\mathbb{Z}$, $r=1,\dots,5$. Under these conditions on C_i, the GWHT is given by

$$\begin{aligned}8|\mathcal{H}_f(u,v,w)|^2=&(2a_1^2+a_2^2+a_3^2+2a_4^2+b_2^2+b_3^2)+2(-1)^w\sin(\varphi_1-\varphi_2)(a_1a_2-a_1b_2)+\\&2(-1)^v\sin(\varphi_1-\varphi_3)(a_1a_3-a_1b_3)+2(-1)^{v\oplus w}\sin(\varphi_2-\varphi_3)(a_3b_2-a_2b_3)+\\&2(-1)^v\sin(\varphi_2-\varphi_4)(a_4b_2-a_2a_4)+2(-1)^w\sin(\varphi_3-\varphi_4)(a_4b_3-a_3a_4)=\\=&(2a_1^2+a_2^2+a_3^2+2a_4^2+b_2^2+b_3^2)+X.\end{aligned}$$

where $a_i=W_{A_i}(u)$, $i=1,\dots,4$, $b_j=W_{A_j+B_j}(u)$, $j=2,3$, $u\in\mathbb{Z}_2^n$. If the equality $X=0$ holds and $a_i^2=b_j^2=1$, then f is a gbent function. Note that $X=0$ induces a condition on the Walsh-Hadamard coefficients of the constituent functions.

To fully specify f, we assume that f is given by

$$f(x,y,z)=\alpha a(x)+\beta b(x)+\gamma c(x)+\delta,$$

and we need to determine α,β,γ and δ from (3.75). Since α,β,γ and δ depend on the variables $(y,z)\in\mathbb{Z}_2^2$ we may consider $Ay+Bz+Cyz+D$ as their general form. Then, the system of equations

$$\begin{aligned}D&=e_{00}\\B+D&=e_{01}\\A+D&=e_{10}\\A+B+C+D&=e_{11},\end{aligned}$$

in variables A, B, C and D, uniquely determines all four functions. The values $\{e_{00}, e_{01}, e_{10}, e_{11}\}$ correspond to the values of the functions, for instance, $e_{00} = \alpha(0,0)$, $e_{01} = \alpha(0,1)$, $e_{10} = \alpha(1,0)$ and $e_{11} = \alpha(1,1)$ (the similar systems apply to functions β, γ and δ). Hence, we obtain

$$\alpha(y,z) = -\frac{q}{4}z + \frac{q}{4}yz + \frac{q}{2}, \;\; \beta(y,z) = \frac{3q}{4}z - \frac{3q}{4}yz, \;\; \gamma(y,z) = \frac{3q}{4}y - \frac{3q}{4}yz,$$

$$\delta(y,z) = (C_3 - C_1)y + (C_2 - C_1)z + (C_1 - C_2 - C_3)yz + C_1.$$

Note that α, β, γ and δ are evaluated modulo q, $q = 4s$, $s \in \mathbb{Z}^+$.

It is not difficult to see that Theorem 22 and Theorem 24 in [111] are just special cases with some particular conditions, as discussed in Section 3.5. Also, Theorem 5 in [107, 108] is a special case of Proposition 3.5.6, Remark 3.5.7.

Chapter 4

Optimizing the placement of tap positions

There are many cryptanalytic approaches that have been applied to nonlinear filter generators during the last two decades. These methods mainly use the cryptographic weaknesses of the filtering function giving rise to Berlekamp-Massey linear complexity attacks [73], linear distinguishing and inversion attacks of Golić [41], [42], [43], algebraic attacks [24], probabilistic algebraic attacks [9], [87], and so on. The basic idea behind the attacks similar to inversion attacks is to exploit the shift of the secret state bits that are used as the input to the filtering function. The designers, well aware of the fact that a proper tap selection plays an important role in the design, mainly use some standard (heuristic) design rationales such as taking the differences between the positions to be prime numbers (if possible), the taps are distributed over the whole LFSR etc.. Intuitively, selecting the taps at some consecutive positions of the LFSR should be avoided (see also [2]), and similarly placing these taps at the positions used for the realization of the feedback connection polynomial is not a good idea either. Even though a full positive difference set is a useful design criterion which ensures that there are no repetitions of several input bits, it is quite insufficient criterion which does not prevent from the attacks such as GFSGA (Generalized Filter State Guessing Attack) introduced in [119]. For instance, assume for simplicity that the inputs to the filtering function are taken at tap positions $\mathcal{I} = \{3, 6, 12, 24\}$ of the employed LFSR, thus our filtering function takes four inputs, i.e., $n = 4$. It is easily verified that all the differences are distinct and the set of (all possible) differences is $D^{\mathcal{I}} = \{i_j - i_k : i_j, i_k \in \mathcal{I}, i_j > i_k\} = \{3, 6, 9, 12, 18, 21\}$. Nevertheless, all these numbers being multiple of 3 would enable an efficient application of GFSGA-like cryptanalysis since the information about the previous states would be maximized. Another criterion considered in the literature, aims at ensuring that a multiset of differences of the tap positions is mutually coprime. This means, that for a given set of tap positions $\mathcal{I} = \{i_1, i_2, \ldots, i_n\}$ of an LFSR of length L (thus $1 \leq i_1 < i_2 < \ldots < i_n \leq L$) all the elements in the difference set $\mathcal{D}^{\mathcal{I}} = \{i_j - i_k : i_j, i_l \in \mathcal{I}, i_j > i_k\}$ are mutually coprime. This condition, which would imply an optimal resistance to GFSGA-like methods, is easily verified to be

impossible to satisfy (taking any two odd numbers their difference being even would prevent from taking even numbers etc.). Therefore, only the condition that the consecutive distances are coprime appears to be reasonable, that is, the elements of $D = \{i_{j+1} - i_j : i_j \in \mathcal{I}\}$ are mutually coprime. An exhaustive search is clearly infeasible, since in real-life applications to select (say) $n = 20$ tap positions for a driving LFSR of length 256 would give $\binom{256}{20} = 2^{98}$ possibilities to test for optimality.

In this chapter, we firstly demonstrate some potentially misleading design rationales from the security point of view and discuss the complexity issues related to optimality (Section 4.1). Indeed, for a standard size of an LFSR used in these schemes, say $L = 256$, and a recommended number of inputs $n \geq 16$, any exhaustive search over the set of $\binom{L}{n}$ elements is clearly infeasible. Therefore, we propose a suboptimal algorithms for this purpose (Section 4.2), which at least when applied to LFSRs of relatively short length performs optimally (giving the best choice over all possibilities). We show that the selection of tap positions in real-life stream ciphers such as SOBER-t32 [46], SFINKS [10] and Grain-128 [1] could have been (slightly) improved to ensure a better resistance of these ciphers to GFSGA-like cryptanalysis (Section 4.2). In particular, the selection of tap positions for Grain-128 cipher is far from being optimized allowing for a significant improvement of its resistance to GFSGA-like attacks as shown in Section 4.5.3. Thus, these algorithms appear to be the only known efficient and generic method for the purpose of selecting tap positions (sub)optimally. In addition, the construction of algorithms is also analyzed in terms of criteria for tap selection proposed in [42] (Section 4.4). It is shown that these criteria are embedded in our algorithms but they are not sufficient for protecting the considered encryption schemes against GFSGA-like methods adequately.

In Section 4.3, we further extend the GFSGA framework by considering a variable mode of sampling which was not addressed in FSGA [88] or GFSGA [119]. The complexity analysis of all modes in terms of the number of repeated state bit equations is addressed here as well. In Section 4.4, the performance of different attack modes and the algorithms for determining a (sub)optimal selection of tap positions are presented. We notice that the main difficulty, when comparing the performance of these modes theoretically, lies in the fact that there are intrinsic trad-offs between the main parameters involved in the complexity computation, cf. Remark 4.3.1. The main reason is that each of these modes attempt to reduce the preimage space based on the knowledge of some secret state bits that reappear as the inputs, but at the same time these linear equations (describing the known/guessed secret state bits) have already been used for setting up a system of linear equations to be solved once the system becomes overdefined. Thus, increasing the number of repeated bits makes a reduction of the preimage space more significant (less bits needs to be guessed) but at the same time more sampling is required since the repeated bits do not increase the rank of the system of linear equations. This is the trade-off that makes the complexity analysis hard and consequently no theoretical results regarding the performance of the attack modes can be given.

Finally, well aware of the main limitation of GFSGA-like attacks, which are efficiently applicable to LFSR-based ciphers with filtering function $F : GF(2)^n \to GF(2)^m$ where $m > 1$, in Section 4.5 we briefly discuss their application to single output filtering functions (thus $m = 1$) and to ciphers employing nonlinear feedback

shift registers (NFSRs). In both cases we indicate that GFSGA attacks may be adjusted to work satisfactory in these scenarios as well. Most notably, there might be a great potential in applying GFSGA attacks in combination with other cryptanalytic techniques such as algebraic attacks. This possibility arises naturally due to the fact that GFSGA-like attacks reduce the preimage space of possible inputs to a filtering function using the knowledge of previous inputs, thus giving rise to the existence of low degree annihilators defined on a restriction of the filtering function (obtained by keeping fixed a subset of known input variables). In another direction, when considering NFSR-based ciphers we propose a novel approach of mounting internal state recovery attacks on these schemes which employs the GFSGA sampling procedure but without solving the deduced systems of equations at all. More precisely, this new type of internal state recovery attack collects the outputs within a certain sampling window which then enables an efficient recovery of a certain portion of internal state bits. This is done by filtering out the wrong candidates based on the knowledge of reduced preimage spaces that correspond to the observed outputs. Note that all results of this chapter are published in [90, 49].

4.1 Complexity versus the number of repeated equations

The complexity of GFSGA, which is a generic attack for this particular encryption scheme, strongly depends on the choice of tap positions, see also [119]. Therefore, our goal is to maximize this complexity which is certainly related to the minimization of the parameters $r_i = \#\mathcal{I}_i$, but not completely equivalent. Notice that by optimizing the resistance of these schemes to GFSGA does not necessarily imply the optimality of tap selections, though for the targeted filtering generator we cannot see other reasonable approaches in the context of the guess and determine cryptanalysis. Using the formulas for complexity computation of the GFSGA attack, given by relations (3.3)-(3.4), in this section we analyze the question whether the sampling step which provides the maximal number of repeated sate bit (equations) imply the minimal attack complexity.

Let R denotes the number of repeated equations regardless of this number being $\sum_{i=1}^{c-1} r_i$ for $c \leq k$, or $\sum_{i=1}^{k} r_i + (c-k-1)r_k$ for $c > k$. From [119], it somehow appears that an (sub)optimal choice of tap positions is the one that minimizes the number of repeated equations R, which is a bit misleading as illustrated by the following example.

Example 4.1.1 *Let the tap positions be given by* $\mathcal{I}_0 = \{1, 5, 13, 25, 41, 65, 77\}$, *for* $L = 80$, $n = 7$, *and* $m = 3$. *Computing the complexity* $T_{Comp.}$ *for all sampling differences* $\sigma = 1, 2, 3, \ldots, 76$, *one can verify that the best choice of* σ *for the attacker is* $\sigma = 12$, *with the minimal complexity* $T_{Comp.} \approx 2^{23.97}$ *and having* $R = 177$ *as the number of repeated equations. However, the computation below shows that for* $\sigma = 4$, $R = 353$ *is maximum possible, but in that case* $T_{Comp.} \approx 2^{27.97}$.

To see why $\sigma = 4$ is not optimal for the attacker, we first compute $r_i = \#\mathcal{I}_i$,

$$\begin{aligned}\mathcal{I}_1 &= \{5\},\ \mathcal{I}_2 = \{5,13\},\ \mathcal{I}_3 = \{5,13,25,77\},\ \mathcal{I}_4 = \{5,13,25,41,77\},\\ \mathcal{I}_5 &= \{5,13,25,41,77\},\ \mathcal{I}_6 = \{5,13,25,41,65,77\},\\ \mathcal{I}_j &= \{5,13,25,41,65,77\},\ \ for\ \ j = 7,8,\ldots,61.\end{aligned}$$

The number of sampling points c, for $k = \lfloor \frac{77-1}{4} \rfloor = 19$, is determined from the condition $nc-(\sum_{i=1}^{k} r_i+(c-k-1)r_k) > L$, i.e., $c = 62$ is the smallest positive integer satisfying the condition. The terms $2^{(n-m-r_i)} \neq 1$ in (3.4), for which $r_i < n-m$ so that the number of preimages is greater than one, only appear for $r_1 = 1$ and $r_2 = 2$, i.e.,

$$T_{Comp.} = 2^{(n-m)} \times 2^{(n-m-r_1)} \times 2^{(n-m-r_2)} \times L^3 \approx 2^{27.97}.$$

For $j = 3,\ldots,61$, we have $2^{(n-m-r_j)} = 1$, in accordance to Remark 2.4.2.

Similarly, for $\sigma = 12$, which implies that $k = 6$, we obtain $c = 37$ (where c is derived from $nc - (\sum_{i=1}^{k} r_i + (c-k-1)r_k) > L$) and "only" $R = 177$ repeated equations. The intersection sets in this case are given as,

$$\begin{aligned}\mathcal{I}_1 &= \{13,25,77\},\ \mathcal{I}_2 = \{13,25,65,77\},\ \mathcal{I}_3 = \{13,25,41,65,77\},\\ \mathcal{I}_j &= \{13,25,41,65,77\},\ \ for\ \ j = 4,5,\ldots,36.\end{aligned}$$

The complexity computation in this case involves only $r_1 = 3$, i.e.,

$$T_{Comp.} = 2^{(n-m)} \times 2^{(n-m-r_1)} \times L^3 \approx 2^{23.97}.$$

Notice that for $j = 2,\ldots,36$, we have $2^{(n-m-r_j)} = 1$.

Remark 4.1.2 *A lower complexity in the above example (for a larger number of repeated equations) is entirely due to a low difference between n and m so that many of the repeated equations could not be efficiently used since the preimages could be identified uniquely even without using these equations.*

More formally, if σ' gives the maximal possible value of R though the attack complexity is not minimal, and σ'' gives the minimal attack complexity without maximizing R, then it holds

$$\sum_{r_j \in H_{\sigma''}} (n-m-r_j) < \sum_{r_i \in H_{\sigma'}} (n-m-r_i) \tag{4.1}$$

where $H_{\sigma'} = \{r_i < n-m\ :\ r_i \text{ obtained by } \sigma',\ \ i = 1,2,\ldots,c-1\}$ and $H_{\sigma''} = \{r_j < n-m\ :\ r_j \text{ obtained by } \sigma'',\ \ j = 1,2,\ldots,c-1\}$. In the above example, we have $H_{\sigma'} = \{r_1, r_2\} = \{1,2\}$ with $\sigma' = 4$, and $H_{\sigma''} = \{r_1\} = \{3\}$ with $\sigma'' = 12$, for which (4.1) holds.

Another problem related to the approach of finding the intersection sets given by (2.5) is that the information contained in R and the cardinalities r_i alone does not fully specifies the properties of the repeated equations. The equations corresponding to the numbers in the sets $\mathcal{I}_i$ may be repeated and found in other sets $\mathcal{I}_j$, where $i \neq j$,

and even though they efficiently reduce the preimage space they do not contribute to the rank of the systems of linear equations that need to be solved. An alternative method of tracking the repeated equations, illustrated in the example bellow, turns out to give a deeper insight to the problem of selecting the tap positions optimally.

Example 4.1.3 *Let the tap positions be given by* $\mathcal{I}_0 = \{l_1, l_2, l_3, l_4, l_5\} = \{1, 4, 8, 9, 11\}$, $L = 15$, *and the sampling distance* $\sigma = 2$. *Let* $\mathbf{s}^{t_i} = (s_{0+(i-1)\sigma}, s_{1+(i-1)\sigma}, \ldots, s_{14+(i-1)\sigma})$, *denote the LFSR state over* $c = 10$ *sampling instances* $t_i = (i-1)\sigma$, *for* $i = 1, 2, \ldots, 10$. *Moreover, at these different sampling instances, we represent the output bits of LFSR* $s_0, s_1, \ldots$ *via their indices in* $\mathbb{N}$, *i.e.,* $s_k \to (k+1) \in \mathbb{N}$. *For instance, in Table 2 the number 27 corresponds to the bit* s_{26} *which becomes a part of the LFSR state* s^{t_9} *at position* l_5. *The LFSR state bits at tap positions* $\mathcal{I}_0 = \{l_1, l_2, l_3, l_4, l_5\}$ *are illustrated in Table 4.1. Our goal is to determine when some equation (state bit)*

Table 4.1: The LFSR state bits at given tap positions for $\sigma = 2$.

States	l_1	l_2	l_3	l_4	l_5
$\mathbf{s}^{t_1}$	$s_0 \to 1$	$s_3 \to 4$	$s_7 \to 8$	$s_8 \to 9$	$s_{10} \to 11$
$\mathbf{s}^{t_2}$	$s_2 \to 3$	$s_5 \to 6$	$s_9 \to 10$	$s_{10} \to 11$	$s_{12} \to 13$
$\mathbf{s}^{t_3}$	$s_4 \to 5$	$s_7 \to 8$	$s_{11} \to 12$	$s_{12} \to 13$	$s_{14} \to 15$
$\mathbf{s}^{t_4}$	$s_6 \to 7$	$s_9 \to 10$	$s_{13} \to 14$	$s_{14} \to 15$	$s_{16} \to 17$
$\mathbf{s}^{t_5}$	$s_8 \to 9$	$s_{11} \to 12$	$s_{15} \to 16$	$s_{16} \to 17$	$s_{18} \to 19$
$\mathbf{s}^{t_6}$	$s_{10} \to 11$	$s_{13} \to 14$	$s_{17} \to 18$	$s_{18} \to 19$	$s_{20} \to 21$
$\mathbf{s}^{t_7}$	$s_{12} \to 13$	$s_{15} \to 16$	$s_{19} \to 20$	$s_{20} \to 21$	$s_{22} \to 23$
$\mathbf{s}^{t_8}$	$s_{14} \to 15$	$s_{17} \to 18$	$s_{21} \to 22$	$s_{22} \to 23$	$s_{24} \to 25$
$\mathbf{s}^{t_9}$	$s_{16} \to 17$	$s_{19} \to 20$	$s_{23} \to 24$	$s_{24} \to 25$	$s_{26} \to 27$
$\mathbf{s}^{t_{10}}$	$s_{18} \to 19$	$s_{21} \to 22$	$s_{25} \to 26$	$s_{26} \to 27$	$s_{28} \to 29$

is repeated on the tap positions $l_1, \ldots, l_4$ *at the sampling instances* t_i. *Hence, we observe the repetition of all consecutive tap positions* $l_{j+1} - l_j$, *then the differences* $l_{j+2} - l_j$, *etc. Let* D *be a set of all differences between consecutive tap positions, i.e.,*

$$D = \{\, d_j \mid d_j = l_{j+1} - l_j, \;\; j = 1, 2, 3, 4\} = \{3, 4, 1, 2\}.$$

To consider all possible repetitions of the equations on all tap positions, we design a scheme of all possible differences: In Table 4.2, Column 1 *specifies the repetition of*

Table 4.2: The scheme of all possible differences for the set D.

Row\Columns	Col. 1	Col. 2	Col. 3	Col. 4
Row 1	d_1	d_2	d_3	d_4
Row 2	$d_1 + d_2$	$d_2 + d_3$	$d_3 + d_4$	
Row 3	$d_1 + d_2 + d_3$	$d_2 + d_3 + d_4$		
Row 4	$d_1 + d_2 + d_3 + d_4$			

some equations at the tap position l_1, *Column* 2 *gives the repetition of equations on* l_2, *etc. Similarly, Row* 1 *takes into account the consecutive repetitions from* l_{i+1} *to* l_i, *Row* 2 *regards the repetition from* l_{i+2} *to* l_i, *etc. In our example, by Table 4.2, we have Assuming the attacker starts the sampling with some step* σ, *the total number*

Table 4.3: The scheme of all differences for $D = \{3, 4, 1, 2\}$.

Row\Columns	Col. 1	Col. 2	Col. 3	Col. 4
Row 1	3	4	1	2
Row 2	7	5	3	
Row 3	8	7		
Row 4	10			

of repeated equations R is the sum of all equations which repeat on each of the tap positions l_j, where $j = 1, 2, 3, 4$.
Since Table 4.2 can be designed for an arbitrary set D, $\#D = n-1$, the repetition of the same equations can be tracked as follows. We are looking for the first number in each column such that it is divisible by σ, which implies that we have the repetition of equations, otherwise there are no repetitions. Notice that in Table 4.3, in Column 1, $\sigma \nmid 3$, which implies that there is no repetition of equations from l_2 at l_1. Also, since $2 \nmid 7$, there is no repetition from l_3 at l_1. However, $2 \mid 8$, which implies that the equation(s) from l_4 will appear on l_1 after $\frac{8}{2} = 4$ sampling instances (cf. Table 4.1 where 9 appears at l_1 when the content of the LFSR is $\mathbf{s}^{t_5}$). Thereafter, one equation from l_4 appears at l_1 for every state $\mathbf{s}^{t_i}$, for $i \geq 5$.
Further, the fact that $2 \mid 8$ and $2 \mid 10$ implies that $2 \mid d_4 = 2$, which means that we have a repetition from l_5 to l_1 at every LFSR state $\mathbf{s}_i^t$, $i \geq 2$. Since Column 1 already contains this number 8 which is divisible by 2, all the repeated equations from l_5 to l_1 are already taken into account, and we do not use number 10 (Table 4.3, Row 4) when calculating the number of repeated equations. So, $\frac{d_4}{2}$ is related to the repetitions of equations from l_5 to l_4.
Hence, the number of repeated equations R, for $c = 10$, is calculated as follows.

1. *On l_1, there are $(c - \frac{d_1+d_2+d3}{\sigma}) = 10 - \frac{8}{2} = 6$ repeated equations.*
2. *On l_2, there are $(c - \frac{d_2}{\sigma}) = 10 - 2 = 8$ repeated equations.*
3. *On l_3, there are NO repeated equations, since we do not have the differences divisible by $\sigma = 2$.*
4. *On l_4, there are $(c - \frac{d_4}{\sigma}) = 9$ repeated equations.*

In total, we have $R = 6 + 8 + 0 + 9 = 23$ repeated equations.

The analysis performed in the above example leads to the following result concerning the number of repeated equations.

Proposition 4.1.4 *Let $\mathcal{I}_0 = \{l_1, l_2, \ldots, l_n\}$ be a set of tap positions, and let*

$$D = \{l_{i+1} - l_i \mid i = 1, 2, \ldots, n-1\} = \{d_1, d_2, \ldots, d_{n-1}\}.$$

The number of repeated equations is calculated as

$$R = \sum_{i=1}^{n-1} (c - \frac{1}{\sigma} \sum_{k=i}^{m} d_k), \tag{4.2}$$

where $\sigma \mid \sum_{k=i}^{m} d_k$ for some $m \in \mathbb{N}$, $i \leq m \leq n-1$ and $\frac{1}{\sigma} \sum_{k=i}^{m} d_k \leq c-1$. Moreover, if $\frac{1}{\sigma} \sum_{k=i}^{m} d_k \geq c$, for some $1 \leq i \leq n-1$, then $(c - \frac{1}{\sigma} \sum_{k=i}^{m} d_k) = 0$. This means that the repetition of the same equations (bits) starts to appear after the LFSR state s^{t_c}.

Remark 4.1.5 *The importance of the above proposition lies in a fact that the counting method of repeated equations does not depend on the relation between the number of sampling points c and k (where $k = \lfloor \frac{i_n - i_1}{\sigma} \rfloor$), i.e., it holds for both $c \leq k$ and $c > k$.*

Notice that, in order to minimize the number of repeated equations, the terms $(c - \frac{1}{\sigma}\sum_{k=i}^{m} d_k)$, $i \leq m \leq n-1$, should be minimized. Hence, we want to avoid the divisibility by σ in the scheme of differences as much as possible. Moreover, for a given length L of LFSR, the differences between $d_i \in D$ should be maximized under the constraint $\sum_{i=1}^{n-1} d_i \leq L - 1$, which is also conditioned by $1 \leq l_1 < l_2 < \ldots < l_n \leq L$. In other words, the goal is to distribute the tap positions over entire LFSR while at the same time keeping the divisibility by σ as low as possible. Clearly, if $\sum_{i=1}^{n-1} d_i = L - 1$, then $l_1 = 1$ and $l_n = L$.

4.2 Two algorithms towards an optimal selection of taps

It turns out that the problem of optimizing the choice of $\mathcal{I}_0$ is closely related to the divisibility of the elements in the corresponding (multi)set of differences D by an arbitrary σ. Thus, instead of searching the set $\mathcal{I}_0$ directly, we focus on the set of differences D. The construction of the set D is however out of reach to be done exhaustively for moderately large L and n, and consequently we use some heuristic techniques to specify D (sub)optimally.

In what follows, we present a method of constructing the set D which gives a low number of repeated equations (confirmed by computer simulations) for every σ. The set D is specified using some heuristic design rationales (see below) and at the same time the differences d_i are maximized.

Step A: Find the elements of the set D. To do this and avoid the divisibility by σ, the following pattern is applied.

1. Prime numbers are the most favourable to join the set D. Since higher values of n dictate the repetitions of some elements in D, the repetition should be kept on minimum with a general tendency to choose co-prime differences. If some even numbers are taken, then the set D should contain just few of them, because they can result in many common (high) factors in the rows of Table 4.2.

2. Maximize the differences d_i under the constraint $\sum_{d_i \in D} d_i \leq L - 1$.

Step B: Find the best ordering of the chosen differences, which basically means that ordering of D is also important. This can be done using the following algorithm with the complexity $O(n! \cdot K)$, where K corresponds to the complexity of calculation $T_{Comp.}$ for all possible σ.

INPUT: The set D and the numbers L, $n = \#D + 1$ and m.
OUTPUT: The best ordering of the chosen differences, that is, an ordered set D that maximizes the complexity of the attack.

STEP 1: Generate a list of all permutations of the elements in D;

STEP 2: For every permutation, find the minimal complexity for all steps σ from 1 to L;
STEP 3: Generate a list of all minimal complexities from Step 2;
STEP 4: Find the maximal value in the list of all minimal complexities;
STEP 5: Return the corresponding permutation of the maximal value.

Open Problem 1 *Find an efficient algorithm, which returns the best ordering of the set D without searching all permutations.*

Remark 4.2.1 *To measure the quality of a chosen set of differences D with respect to the maximization of $T_{Comp.}$ over all σ, the computer simulations indicate that an optimal ordering of the set D implies a small value of an optimal sampling distance σ. This is also a criterion that a set D is most likely chosen well (a sub-optimal choice). The term "most likely" concerns the difficulties of capturing the whole process of choosing the tap positions explicitly, due to a very complicated relation between σ, R, D and $T_{Comp.}$ through the scheme of differences. When choosing an output permutation (cf. Step 5 below), we always consider both σ and $T_{Comp.}$ though σ turns out to be a more stable indicator of the quality of a chosen set D.*

Note that, the above algorithm performs an exhaustive search over all permutations of the input set. For practical values of L, usually taken to be $L = 256$, the time complexity of the above algorithm becomes practically infeasible already for $n > 10$. To reduce its factorial time complexity, we modify the above algorithm to process the subsets of the multiset D separately within the feasibility constraints imposed on the cardinalities of these subsets.

STEP 1: Choose a set X by **Step A**, where $\#X < \#D$ for which **Step B** is feasible;
STEP 2: Find the best ordering of X using the algorithm in **Step B** for $L_X = 1 + \sum_{x_i \in X} x_i < L$ and $m_X = \lfloor \#X \cdot \frac{m}{n-1} \rfloor$;
STEP 3: Choose a set Y by **Step A**, where $\#Y < \#D$ for which **Step B** is feasible;
STEP 4: "Generate" a list of all permutations of the elements in Y;
STEP 5: Find a permutation (Y_p) from the above list such that for a fixed set X, the new set Y_pX obtained by joining X to Y_p, denoted by Y_pX (with the parameters $L_{Y_pX} = 1 + \sum_{x_i \in X} x_i + \sum_{y_i \in Y_p} y_i \leq L$ and $m_{Y_pX} = \lfloor \#Y_pX \cdot \frac{m}{n-1} \rfloor$), allows a small optimal step σ, in the sense of Remark 4.2.1;
STEP 6: If such a permutation, resulting in a small value of σ, does not exist in Step 5, then back to Step 3 and choose another set Y;
STEP 7: Update the set $X \leftarrow Y_pX$, and repeat the steps 3 - 5 by adjoining new sets Y_p until $\#Y_pX = n - 1$;
STEP 8: Return the set $D = Y_pX$.

Remark 4.2.2 *The parameters L_X and m_X are derived by computer simulations, where L_X essentially constrains the set X and m_X keeps the proportionality between the numbers $m, \#X$ and $\#D = n - 1$.*

An illustration of our modified version of the above algorithm is given in the following example. Namely, for a rather practical choice of the parameters L, n and m, the whole procedure of defining the set of differences that eventually yields the tap positions is discussed. Some suboptimal choices of tap positions for varying input parameters L, n, m along with the time complexity of the GFSGA and the time complexity of applying our algorithms are given in Appendix (cf. Table 4.4 and Table 4.5).

Example 4.2.3 *Let $n = 17$, $m = 6$, and $F(x) : GF(2)^{17} \longrightarrow GF(2)^6$. Let $L = 160$ bits, the length of the secret key is $K = 80$ bits.*

Let $X = \{5, 13, 7, 26, 11, 17\}$ be obtained using the algorithm in **Step B** *for $L_X = 80$, $m_X = 2$. Let $Y = \{1, 2, 9, 15, 23\}$. Then, a permutation $Y_p = \{9, 1, 2, 23, 15\}$, i.e., the set*

$$Y_pX = \{9, 1, 2, 23, 15, 5, 13, 7, 26, 11, 17\},$$

where $L_{Y_pX} = 130$ and $m_{Y_pX} = 4$, gives that $\sigma = 1$ is an optimal sampling distance for the attacker. Since $L_{Y_pX} \leq 160$, then we choose the set $Z = \{3, 4, 5, 7, 11\}$. Then, a permutation $Z_p = \{5, 11, 4, 3, 7\}$, i.e., the set $Z_pY_pX = \{5, 11, 4, 3, 7, 9, 1, 2, 23, 15, 5, 13, 7, 26, 11, 17\}$, where $L_{Z_pY_pX} = L = 160$ and $m_{L_{Z_pY_pX}} = m = 6$ gives the optimal step $\sigma = 1$ for the attacker. Then we have

$$D = \{5, 11, 4, 3, 7, 9, 1, 2, 23, 15, 5, 13, 7, 26, 11, 17\},$$

and thus

$$\mathcal{I}_0 = \{1, 6, 17, 21, 24, 31, 40, 41, 43, 66, 81, 86, 99, 106, 132, 143, 160\}.$$

Hence, $\sigma = 1$ is optimal, with the minimal complexity $T_{Comp.} = 2^{86.97}$, which is essentially an extremely good choice of tap positions (non-exhaustively confirmed to be an optimal choice).

In Table 4.4 we give several instances for determining suboptimal tap positions of LFSRs of different length. The following parameters are used:

- L is the length of LFSR;
- n and m are parameters related to vectorial Boolean function $F : GF(2)^n \to GF(2)^m$;
- D is a set of differences between tap positions;
- c is the minimal number of observed outputs needed for an overdefined system
- R is the number of repeated equations for given c outputs;
- σ is an optimal step of the GFSGA attack;
- $T_{Comp.}$ is the time complexity of GFSGA.

Remark 4.2.4 *From the difference sets D in Table 4.4 we easily obtain the tap positions.*

Table 4.4: Specifications of difference sets for LFSRs of different lengths.

L	(n,m)	D	R	c	σ	$T_{Comp.}$
80	(7,2)	{5, 13, 7, 26, 11, 17}	24	15	1	$2^{69.97}$
120	(13,3)	{5, 7, 3, 13, 6, 11, 5, 11, 7, 13, 21, 17}	61	14	3	$2^{99.7}$
160	(17,6)	{5,11,4,3,7,9,1,2,23,15,5, 13, 7, 26, 11, 17}	128	17	1	$2^{86.97}$
200	(21,7)	{3, 7, 9, 13, 18, 7, 9, 1, 2, 9, 1, 2, 23, 15, 5, 13, 7, 26, 11, 17}	175	18	1	$2^{108.9}$
256	(27,9)	{5, 9, 13, 4, 7, 19, 3, 7, 9, 13, 18, 7, 9, 1, 2, 9, 1, 2, 23, 15, 5, 13, 7, 26, 11, 17}	227	18	1	2^{135}

Table 4.5: Time complexities for finding tap positions in Table 5.

L	(n,m)	Cardinality of parts	Complexity	Times in *sec*
80	(7,2)	no parts	$O(K \cdot 6!)$	135
120	(13,3)	(6,6)	$2 \cdot O(K \cdot 6!)$	125+162=287
160	(17,6)	(6,6,4)	$2 \cdot O(K \cdot 6!) + O(K \cdot 4!)$	137+198+8.5=343.5
200	(21,7)	(6,6,4,4)	$2 \cdot O(K \cdot 6!) + 2 \cdot O(K \cdot 4!)$	137+96+7.7+9.5=250
256	(27,9)	(6,6,4,4,6)	$3 \cdot O(K \cdot 6!) + 2 \cdot O(K \cdot 4!)$	250+369.3=619.3

Remark 4.2.5 *Note that the time required to create some particular set of differences depends on the cardinality of parts. It means that the smaller cardinalities implies the lower time complexity, though such an approach may provide the solutions that are "far" from optimal. Table 4.5 presents the following:*

- *Cardinality of parts refers to the modified algorithm on Page 10, bottom. For instance,* $(6,6,4)$ *means that we take* $\#X = 6$ *elements and finding its optimal permutation requires* 137 *sec with our permutation algorithm. Then, we take another* $\#Y_p = 6$ *elements and determine its best order which fits to the set* X*, which requires* 198 *seconds (modified algorithm). Finally, the same procedure is applied to the set* Y_pX *by adding* $Z_p = 4$ *elements using again our modified algorithm (requiring 8.5 sec). The resulting set of differences is given as* $D = Z_pY_pX$*.*
- *Complexity refers to the complexity of the permutation algorithms* ***Step B*** *and its modification used to construct the set* D*.*
- *The constant* K *regards the procedure described in the permutation algorithm (****Step B****): creating the list, searching, etc.*

In what follows, we apply the above algorithms to two well-known stream cipher SOBER-t32 [46], [119] and SFINX [10].

SOBER-t32: An application of the GFSGA attack on unstuttered SOBER-t32 was considered in [119]. The tap positions of SOBER-t32 are given by $\mathcal{I}_0 = \{1, 4, 11, 16, 17\}$ (corresponding to the reverse order of the taps $1 \leftarrow s_{16}$, $4 \leftarrow s_{13}$, etc.) and the sampling distance used in [119] was $\sigma = 3$. Due to the reverse order of the bits s_i, we consider the set D in reverse order , i.e. $D = \{1, 5, 7, 3\}$ instead of $\{3, 7, 5, 1\}$, since this ordering corresponds to our consideration of the

LFSR states presented in Table 4.1. Regarding the set D, the set of all $r_i = \#\mathcal{I}_0$ is $\{1, 1, 2, 2, 3, 3, 4, 4, 4, 4, \ldots\}$, i.e. $r_1 = r_2 = 1$, $r_2 = r_3 = 2$, $r_4 = r_5 = 3$ and $r_k = 4$, $k \geq 7$. At each sampling point we derive $40 - 8 \times r_i$ linear equations (cf. [119]). Therefore, the number of repeated equations is given by

$$40 + 32 + 32 + 24 + 24 + 16 + 16 + 8 \times (c - 7) + c, \tag{4.3}$$

which for $c = 47$ gives $R = 550$ linear equations (4.3). Thus the complexity of the attack can be estimated as

$$T_D = (17 \times 32)^3 \times 2^{35} \times 2^{2\times 27} \times 2^{2\times 19} \times 2^{2\times 11} \times 2^{39\times 3} = (17 \times 32)^3 \times 2^{266}.$$

Since $\#D = 4$, we can easily apply **Step A** and **Step B**, to come up with the new set $D^* = \{5, 2, 7, 2\}$, and get the set $\{0, 2, 2, 2, 3, 3, 4, 4, 4, 4, \ldots\}$ of all $r_i = \#\mathcal{I}_0$. The inequality

$$40 + 40 + 32 + 32 + 32 + 24 + 24 + 8 \times (c - 7) + c \geq 544$$

implies $c = 42$, and $R = 546$ equations. The complexity is estimated as

$$T_{D^*} = (17 \times 32)^3 \times 2^{2\times 35} \times 2^{3\times 27} \times 2^{2\times 19} \times 2^{34\times 3} = (17 \times 32)^3 \times 2^{291}.$$

This means that our algorithm gives the tap selection with much better resistance against GFSGA.

SFINX: The design details of SFINX can be found in [10]. The set of the tap positions of SFINX is given as

$$\mathcal{I}_0 = \{1, 2, 7, 10, 20, 22, 45, 59, 75, 99, 106, 135, 162, 194, 228, 245, 256\},$$

and $D = \{1, 5, 3, 10, 2, 23, 14, 16, 24, 7, 29, 27, 32, 34, 17, 11\}$. An optimal step of the GFSGA attack on this set of tap positions, is $\sigma = 2$ which requires $c = 27$ sampling points, resulting in $R = 200$ sampled equations for obtaining an overdefined system. The corresponding complexity in this case is $T_{Comp.} = 2^{256}$. Note that $\sum_{i=1}^{16} d_i = 255$ with optimal step $\sigma = 2$, which indicates that the set of tap positions $\mathcal{I}_0$ of SFINX is chosen well. However, we can use the elements of the given set D and our algorithm to create the set of differences "by parts", in order to decrease the number of repeated equations R and increase the complexity (slightly). Starting with the set $X = \{29, 32, 17, 34, 27, 11\}$, and permuting the set $Y_p = \{2, 23, 14, 16, 24, 7\}$ for $L_{Y_pX} = 237$, we get the set $Y_pX = \{2, 23, 14, 7, 16, 24, 29, 32, 17, 34, 27, 11\}$ with an optimal step $\sigma = 8$ for the attack. Then, taking the set $Z_p = \{1, 5, 3, 10\}$, we get the set $D^* = Z_pY_pX$ given as

$$D^* = \{1, 5, 3, 10, 2, 23, 14, 7, 16, 24, 29, 32, 17, 34, 27, 11\},$$

with the optimal steps $\sigma \in \{1, 2\}$ for the attack. The estimated complexity for both optimal steps is $T_{Comp.} = 2^{257}$ with $R = 167$ repeated equations, thus only a minor improvement has been achieved.

It would be of interest to consider the problem of optimizing the placement of tap positions in case the GFSGA attack with a variable sampling step (σ is not fixed) is used, which is left for the extended version of this article.

4.3 GFSGA with a variable sampling step

In this section, we describe the GFSGA method with a variable sampling step σ, which we denote as $GFSGA^*$. The whole approach is quite similar to $GFSGA$, the main difference is that we consider outputs at any sampling distances, i.e., the observed outputs $z^{t_{i-1}}$ and z^{t_i} $(i = 1, \ldots, c)$ do not necessarily differ by a fixed constant value and consequently the sampling distances $\sigma_1, \ldots, \sigma_c$ are not necessarily the same. It turns out that this approach may give a significant reduction in complexity compared to the standard version of the attack, see Section 4.4.

We first adopt some notation to distinguish between the two modes. The number of observed outputs for which an overdefined system is obtained is denoted by c^*, the corresponding number of repeated bits by R^* and the attack complexity by $T^*_{Comp.}$. The outputs taken at time instances t_i are denoted by w^{t_i}, where we use the variable sampling steps σ_i so that $t_{i+1} = t_i + \sigma_i$, for $i = 1, 2, \ldots, c^* - 1$. These values σ_i (distances between the sampled outputs), are referred to as the variable steps (distances). Throughout this article, for easier identification of repeated bits over observed LFSR states, the state bits $s_0, s_1, \ldots$ are represented via their indices in $\mathbb{N}$, i.e., $s_i \to (i+1) \in \mathbb{N}$ $(i \geq 0)$. In other words, the LFSR state bits $s^i = (s_{0+i}, s_{1+i}, \ldots, s_{L-1+i})$ are treated as a set of integers given by

$$(s_{0+i}, s_{1+i}, \ldots, s_{L-1+i}) \leftrightarrow \{1+i, 2+i, \ldots, L+i\}. \tag{4.4}$$

The purpose of this notation is to simplify the formal definition of LFSR state bits at tap positions introduced in the previous section. In addition, it allows us to easier track these bits and to count the number of repeated bits using the standard concepts of union or intersection between the sets. Henceforth, the LFSR states s^i we symbolically write as $s^i = \{1+i, 2+i, \ldots, L+i\}$ $(i \geq 0)$, and this notation applies to state bits at tap positions s^{t_i}. Since finding the preimage spaces $S_{w^{t_i}}$ of the observed outputs w^{t_i} is the most important part, we give for clarity the description of a few initial steps:

Step 1: Let w^{t_1} denotes the first observed output so that the corresponding LFSR state at the tap positions is exactly the set $\mathcal{I}_0 = \{l_1, l_2, \ldots, l_n\}$, so that $s^{t_1} = (s^{t_1}_{l_1}, \ldots, s^{t_1}_{l_n}) \overset{(4.4)}{=} \{l_1, \ldots, l_n\}$, i.e., $s^{t_1} = \mathcal{I}_0$. Notice that the first observed output w^{t_1} does not necessarily need to correspond to the set $\mathcal{I}_0$, though (for simplicity) we assume this is the case. A preimage space which corresponds to the first observed output w^{t_1} always has the size 2^{n-m}, i.e., $|S_{w^{t_1}}| = 2^{n-m}$.

Step 2: Taking the second output w^{t_2} at distance σ_1 from w^{t_1} (thus $t_2 = t_1 + \sigma_1$), we are able to identify and calculate the number of repeated bits (equations) at the time instance t_2. Using the notation above, the set $\mathcal{I}^*_1$ of these bits is given by the intersection:

$$\mathcal{I}^*_1 = s^{t_1} \cap s^{t_2} = s^{t_1} \cap \{s^{t_1} + \sigma_1\} = \mathcal{I}_0 \cap \{l_1 + \sigma_1, \ldots, l_n + \sigma_1\},$$

where $s^{t_2} = \{s^{t_1} + \sigma_1\} \overset{\text{def}}{=} (s^{t_1+\sigma_1}_{l_1+\sigma_1}, s^{t_2+\sigma_1}_{l_2+\sigma_1}, \ldots, s^{t_n+\sigma_1}_{l_n+\sigma_1}) \overset{(4.4)}{=} \{l_1 + \sigma_1, \ldots, l_n + \sigma\}$, i.e., $s^{t_2} = \{l_1 + \sigma_1, \ldots, l_n + \sigma\}$ is the LFSR state at tap positions at time instance t_2.

Denoting the cardinality of $\mathcal{I}_1^* = s^{t_1} \cap s^{t_2}$ by q_1, i.e., $q_1 = \#\mathcal{I}_1^*$, the cardinality of the preimage space corresponding to the output w^{t_2} is given as $|S_{w^{t_2}}| = 2^{n-m-q_1}$.

This process is then continued using the sampling distances $\sigma_2, \ldots, \sigma_{c^*}$ until the condition $nc^* - R^* > L$ is satisfied, where the total number of repeated equations over c^* observed outputs is $R^* = \sum_{k=1}^{c^*-1} q_k$. Note that the number of repeated equations corresponding to the first output is 0, since the corresponding LFSR state s^{t_1} is the starting one. Therefore the sum goes to $c^* - 1$.

It is not difficult to see that the equalities (2.5), used in $GFSGA$ to determine the parameters $r_i = \#\mathcal{I}_i$, are special case of the equalities given as:

$$\begin{aligned}
\mathcal{I}_1^* &= \mathcal{I}_0 \cap \{l_1 + \sigma_1, l_2 + \sigma_1, \ldots, l_n + \sigma_1\} = s^{t_1} \cap s^{t_2}, \\
\mathcal{I}_2^* &= \{s^{t_1} \cup s^{t_2}\} \cap \{l_1 + (\sigma_1 + \sigma_2), \ldots, l_n + (\sigma_1 + \sigma_2)\} = \{s^{t_1} \cup s^{t_2}\} \cap s^{t_3}, \\
&\vdots \\
\mathcal{I}_j^* &= \{s^{t_1} \cup \ldots \cup s^{t_j}\} \cap \{l_1 + \sum_{i=1}^{j} \sigma_i, \ldots, l_n + \sum_{i=1}^{j} \sigma_i\} = \bigcup_{i=1}^{j} s^{t_i} \cap s^{t_{j+1}}, \\
&\vdots \\
\mathcal{I}_{c^*-1}^* &= \bigcup_{i=1}^{c^*-1} s^{t_i} \cap s^{t_{c^*}},
\end{aligned} \tag{4.5}$$

where $s^{t_1}, \ldots, s^{t_{c^*}}$ represents the LFSR state bits at tap positions at time instances $t_1, \ldots, t_{c^*}$.

In general, the number of repeated equations which corresponds to the outputs $w^{t_1}, \ldots, w^{t_{c^*}}$ at variable distances σ_i (i.e., $w^{t_{i+1}} = w^{t_i + \sigma_i}$), can be calculated as

$$q_j = \#\mathcal{I}_j^* = \# \left\{ \bigcup_{i=1}^{j} s^{t_i} \cap \{s^{t_1} + \sum_{i=1}^{j} \sigma_i\} \right\}, \tag{4.6}$$

where all steps σ_i are fixed for $i = 1, \ldots, j$ and $j = 1, \ldots, c^* - 1$. Note that the sets $\mathcal{I}_{j-1}^*$ $(j \geq 1)$ correspond to outputs w^{t_j} (where $\mathcal{I}_0^* = \mathcal{I}_0 \overset{(4.4)}{=} s^{t_1}$). The sampling instances are given as $t_j = t_1 + \sum_{i=1}^{j-1} \sigma_i$, or $t_j = t_{j-1} + \sigma_{j-1}$, where $t_{j-1} = t_1 + \sum_{i=1}^{j-2} \sigma_i$ is fixed. Similarly to the GFSGA with a constant sampling distance, the attack complexity is estimated as

$$T_{Comp.}^* = 2^{n-m} \times 2^{n-m-q_1} \times \ldots \times 2^{n-m-q_{c^*-1}} \times L^3. \tag{4.7}$$

Remark 2.4.2 also applies here, thus if $n - m - q_j \leq 0$ for some $j \in \{1, \ldots, c^* - 1\}$, then the knowledge of these q_j bits allows the attacker to uniquely identify the exact preimage value of the observed output, i.e., we have $2^{(n-m-q_j)} = 1$ when $n - m - q_j \leq 0$. Notice that if the sampling steps σ_i are equal, i.e., they have a constant value $\sigma = \sigma_i$, for $i = 1, 2, \ldots, c^* - 1$, we get $q_i = r_i$, $c^* = c$, $R^* = R$ and $T_{Comp.}^* = T_{Comp.}$.

Remark 4.3.1 *It was already mentioned that the analysis of complexity $T_{Comp.}^*$ appears to be very difficult, mainly due to the following reasons. For fixed m, n and*

L, it is clear that there exists a trade-off between the parameters q_j ($j = 1, \ldots, c^* - 1$) and c^*. More precisely, for larger values c^* we have that $2^{n-m-q_j} > 1$ which implies the increase of $T^*_{Comp.}$, unless $n - m - q_j \leq 0$. For this reason, the optimal step(s) of the GFSGA attack (whether we consider a constant or variable mode of the attack) is the one which minimizes c^* satisfying at the same time the inequality $nc^* - R^* > L$. Furthermore, in the case of the constant GFSGA mode, the parameter c for which $nc - R > L$ holds is not known prior to the completion of the sampling process. This also holds for the variable GFSGA mode, if we fix a sequence of sampling steps in advance. This, in combination with [90, Remark 3], give more insight how complicated the relation between parameters m, n, L, q_j, c^* and $T^*_{Comp.}$ is.

4.3.1 The number of repeated equations for $GFSGA^*$

The relation between the number of repeated equations and complexity in the case of $GFSGA$ has been analyzed in Section 4.1, where an alternative method for calculating the number of repeated equations has been derived. Similarly, in this section we derive an alternative method for calculating the number of repeated equations for $GFSGA^*$ (Proposition 4.3.3).

For a given set of tap positions $\mathcal{I}_0 = \{l_1, l_2, \ldots, l_n\}$, let us consider the set of differences between the consecutive tap positions, i.e.,

$$D = \{\, d_j \mid d_j = l_{j+1} - l_j, \;\; j = 1, 2, \ldots, n-1\}.$$

Based on this set the so-called scheme of all possible differences was defined in Section 4.1 as $D^{\mathcal{I}_0} = \{l_j - l_k : l_j, l_k \in \mathcal{I}_0, l_j > l_k\}$ and used to calculate the number of repeated equations for GFSGA. For self-completeness we recall Proposition from Section 4.1.

Proposition 4.3.2 *Let $\mathcal{I}_0 = \{l_1, l_2, \ldots, l_n\}$ be a set of tap positions, and let*

$$D = \{l_{j+1} - l_j \mid j = 1, 2, \ldots, n-1\} = \{d_1, d_2, \ldots, d_{n-1}\}.$$

The number of repeated equations is calculated as

$$R = \sum_{i=1}^{n-1} (c - \frac{1}{\sigma} \sum_{k=i}^{m} d_k), \tag{4.8}$$

where $\sigma \mid \sum_{k=i}^{m} d_k$ for some $m \in \mathbb{N}$, $i \leq m \leq n-1$ and $\frac{1}{\sigma} \sum_{k=i}^{m} d_k \leq c-1$. Moreover, if $\frac{1}{\sigma} \sum_{k=i}^{m} d_k \geq c$, for some $1 \leq i \leq n-1$, then $(c - \frac{1}{\sigma} \sum_{k=i}^{m} d_k) = 0$. This means that the repetition of the same equations (bits) starts to appear after the LFSR state s^{t_c}.

In the case of $GFSGA^*$, the scheme of differences can also be used to calculate the number of repeated equations R^*. However, in this case the calculation is slightly more complicated compared to GFSGA, due to the fact that we have a variable step of sampling.

To illustrate the difference, let us consider the set of tap positions given by $\mathcal{I}_0 = \{3, 5, 10, 14, 16\}$ ($L = 20$ and n, m not specified). The corresponding set of

consecutive differences is given as $D = \{2, 5, 4, 2\}$. The scheme of all differences related to $D^{\mathcal{I}_0}$ is given as:

Table 4.6: The scheme of all differences for $D = \{2, 5, 4, 2\}$.

	Col. 1	Col. 2	Col. 3	Col. 4
$l_{j+1} - l_j$	2	5	4	2
$l_{j+2} - l_j$	7	9	6	
$l_{j+3} - l_j$	11	11		
$l_{j+4} - l_j$	13			

In addition, let us assume that the first two steps of sampling (distances between observed outputs) are given as: $\sigma_1 = 5$, $\sigma_2 = 2$. To find the number of repeated bits (equations), we use the recursion of the sets $\mathcal{I}_k^*$ given by relation (4.5). Even though c^* is the number of outputs for which an overdefined system can be set up, our purpose is to demonstrate the procedure of finding repeated bits for σ_1, σ_2. The computation of the number of repeated bits is as follows:
1) The state bits at tap positions at time t_1 correspond to $\mathcal{I}_0 = \{l_1, l_2, l_3, l_4, l_5\} = \{3, 5, 10, 14, 16\}$, thus $s^{t_1} = (s_2, s_4, s_9, s_{13}, s_{15}) \overset{(4.4)}{=} \mathcal{I}_0$. Since the first sampling distance $\sigma_1 = 5$, we consider the LFSR state $s^{t_2} = \{\mathcal{I}_0 + \sigma_1\}$ which is given as

$$s^{t_2} = \{\mathcal{I}_0 + 5\} = (s_7, s_9, s_{14}, s_{18}, s_{20}) \overset{(4.4)}{=} \{8, 10, 15, 19, 21\}.$$

We obtain that $\mathcal{I}_1^* = s^{t_1} \cap s^{t_2} = \mathcal{I}_0 \cap (\mathcal{I}_0 + 5) = \{10\}$, which means that $1 = \#\mathcal{I}_1^* = q_1$ bit is repeated and found in s^{t_2} from the first state $s^{t_1} = \mathcal{I}_0$.

In terms of the scheme of differences, this repetition corresponds to $d_2 = l_3 - l_2 = \sigma_1 = 5$, found as the first entry in Col. 2. In addition, note that $d_2 = 5$ is the only entry in the scheme of differences $D^{\mathcal{I}_0}$ which is equal to σ_1. The main difference compared to GFSGA is that in this case we do NOT consider the divisibility by σ_1 in the scheme of differences due to variable sampling steps.

2) For $\sigma_2 = 2$, the observed outputs w^{t_2} and w^{t_3} satisfy $w^{t_3} = w^{t_2+\sigma_2} = w^{t_1+\sigma_1+\sigma_2}$. The LFSR state s^{t_3}, which corresponds to the output w^{t_3}, is given as $s^{t_3} = \{s^{t_2} + \sigma_2\} = \{s^{t_1} + (\sigma_1 + \sigma_2)\}$, and therefore

$$s^{t_3} = \{s^{t_2} + 2\} = (s_9, s_{11}, s_{16}, s_{20}, s_{22}) = \{10, 12, 17, 21, 23\}.$$

At this position we check whether there are repeated bits from the LFSR state s^{t_2}, but also from the state s^{t_1}. To find all bits which have been repeated from the state s^{t_2}, we consider the intersection $s^{t_3} \cap s^{t_2} = \{10, 21\}$. In addition, the bits which are repeated from the state s^{t_1} are given by the intersection $s^{t_3} \cap s^{t_1} = \{10\}$. Hence, we have the case that the same bit, indexed by 10, has been shifted from the state s^{t_1} (since $\sigma_1 + \sigma_2 = 7$ so that 3 + 7=10) and from s^{t_2} (since $\sigma_2 = 2$) to s^{t_3}. On the other hand, the intersection corresponding to 21 gives us an equation that has not been used previously. Thus, the number of known state bits used in the reduction of the preimage space is 2, and therefore $|S_{w^{t_3}}| = 2^{n-m-2}$. In terms of the scheme of differences, one may notice that we have $d_1 = l_2 - l_1 = 2 = \sigma_2$ (which refers to repetition from s^{t_2} to s^{t_3}) and which gives $d_1 = 2$ in Col. 1 and Col. 4. On the other hand, for $d_1 + d_2 = l_3 - l_1 = 7 = \sigma_1 + \sigma_2$ (which refers to repetition from s^{t_1} to s^{t_3})

the repeated bit is found in the same column, namely Col. 1, and therefore it is not counted. In general, we conclude here that if we have a matching of $\sigma^{(2)} = d_2$ and $\sigma^{(1)} = d_1 + d_2$ with some numbers (entries) in the scheme of differences which are in the same column (as we have here $d_1 = \sigma^{(2)}$ and $d_1 + d_2 = \sigma^{(1)}$), we calculate only one repeated bit in total from this column. If there were more than 2 matchings, the same reasoning applies and thus we would calculate only one repeated bit.

The procedure above may continue for any number of observed outputs at any sampling distances. For instance, if we consider some sampling step σ_j ($j \geq 1$), then we also need to consider all sums of the steps $\sigma^{(j-i)} = \sum_{h=j-i}^{j} \sigma_h$, for all $i = 0, \ldots, j-1$ ($j \geq 1$) and their matchings with some entries in the scheme of differences. In general, every repeated bit means that some sum(s) of steps $\sigma^{(j-i)} = \sum_{h=j-i}^{j} \sigma_h$, $i = 0, \ldots, j-1$ is equal to some difference $\sum_{p=r}^{m} d_p$, for some $m \in \mathbb{N}$, over different columns, where $r = 1, \ldots, n-1$ relate to the columns in the scheme of differences. A total number of repeated bits R^* is the sum of all repeated bits over observed outputs at distances σ_j ($j \geq 1$). Note that the method for calculation of the number of repeated bits described above actually generalizes Proposition 4.3.2, since the use of constant sampling distance is just a special case of variable sampling.

Proposition 4.3.3 *Let $\mathcal{I}_0 = \{l_1, l_2, \ldots, l_n\}$ be a set of tap positions, and let*

$$D = \{l_{i+1} - l_i \mid i = 1, 2, \ldots, n-1\} = \{d_1, d_2, \ldots, d_{n-1}\}.$$

Denoting $\sigma^{(j-i)} = \sum_{h=j-i}^{j} \sigma_h$, $i = 0, \ldots, j-1$, the number of repeated equations is calculated as

$$R^* = \sum_{j=1}^{c^*-1} q_j = \sum_{j=1}^{c^*-1} \left(\sum_{i=0}^{j-1} \frac{1}{\sigma^{(j-i)}} \sum_{p=r}^{m} d_p \right), \tag{4.9}$$

where the term $\frac{1}{\sigma^{(j-i)}} \sum_{p=r}^{m} d_p = 1$ if and only if $\sigma^{(j-i)} = \sum_{p=r}^{m} d_p$ for some $m, r \in \mathbb{N}$, $1 \leq r \leq m \leq n-1$, otherwise it equals 0. If for a fixed r and different numbers m we have more matchings $\sigma^{(j-i)} = \sum_{p=r}^{m} d_p$, then only one bit will be taken in calculation.

Clearly, the numbers m and r depend on the values $\sigma^{(j-i)}$, $i = 0, \ldots, j-1$ ($j \geq 1$) since we only consider those numbers $m, r \in \mathbb{N}$ such that $\sum_{p=r}^{m} d_p$ is equal to $\sigma^{(j-i)}$.

4.3.2 Two specific modes of $GFSGA^*$

In this section we present two modes of $GFSGA^*$, which in comparison to $GFSGA$ depend less on the choice of tap positions. First we start with a general discussion regarding the attack complexity.

In order to obtain a minimal complexity of the $GFSGA^*$ attack, it turns out that the main problem is actually a selection of the cipher outputs. This problem is clearly equivalent to the problem of selecting the steps σ_i which gives the minimal complexity $T^*_{Comp.}$. The number of repeated bits (equations) at time instance t_i (for some $i > 1$) always depends on all previous sampling points at $t_1, \ldots, t_{i-1}$.

This property directly follows from (4.5), i.e., from the fact that we always check the repeated bits which come from the tap positions of LFSR at time instances $t_1, \ldots, t_{i-1}$. This means that the number of repeated bits q_i at time instances t_i, given by (4.6), always depends on the previously chosen steps $\sigma_1, \ldots, \sigma_{i-1}$. This also implies that we cannot immediately calculate the number of required keystream blocks c^* for which the inequality $nc^* - R^* > L$ is satisfied. This inequality can only be verified subsequently, once the sampling distances and the number of outputs c^* have been specified. Therefore we pose the following problem.

Open Problem 2 *For a given set of tap positions $\mathcal{I}_0 = \{l_1, \ldots, l_n\}$, without the knowledge of c^*, determine an optimal sequence of sampling distances σ_i for which the minimal complexity $T^*_{Comp.}$ is achieved.*

In what follows we provide two modes of the $GFSGA^*$ whose performance will be discussed in Section 4.4.

4.3.3 $GFSGA^*_{(1)}$ mode of attack

In order to minimize the complexity $T^*_{Comp.}$, one possibility is to maximize the values q_j, given by (4.6), by choosing suitable σ_i. However, this approach implies a trade-off between the values q_j and c^*, since c^* is not necessarily minimized. More precisely:
1) For the first step $1 \leq \sigma_1 \leq L$ we take a value for which q_1 is maximized, i.e., for which the cardinality

$$q_1 = \#\{s^{t_1} \cap (s^{t_1} + \sigma_1)\} = \#\{s^{t_1} \cap s^{t_2}\}$$

is maximized. Without loss of generality, we can take the minimal σ_1 for which this holds.
2) In the same way, in the second step we take a value $1 \leq \sigma_2 \leq L$ for which

$$q_2 = \#\{(s^{t_1} \cup s^{t_2}) \cap s^{t_3}\} = \#\{(s^{t_1} \cup s^{t_2}) \cap ((s^{t_1} + \sigma_1) + \sigma_2)\}$$

is maximized. As we know, the step σ_1 here is fixed by the previous step. We continue this procedure until an overdefined system is obtained.

In other words, the values q_j are determined by the *maximum* function over σ_i, for $1 \leq \sigma_i \leq L$, i.e., we choose the steps σ_i for which we have:

$$q_j = \max_{1 \leq \sigma_j \leq L} \# \left\{ \bigcup_{i=1}^{j} s^{t_i} \cap \{(s^{t_1} + \sum_{i=1}^{j-1} \sigma_i) + \sigma_j\} \right\}, \tag{4.10}$$

where $\sum_{i=1}^{j-1} \sigma_i$ is fixed and $j = 1, \ldots, c^* - 1$. Hence, the function $\max_{1 \leq \sigma_j \leq L}$ used in (4.10) means that we are choosing σ_i so that the maximal intersection of $s^{t_{j+1}}$ with all previous LFSR states $s^{t_1}, \ldots, s^{t_j}$ (in terms of cardinality) is achieved. This mode of $GFSGA^*$ we denote by $GFSGA^*_{(1)}$.

4.3.4 $GFSGA^*_{(2)}$ mode of attack

Another mode of $GFSGA^*$, based on the use of sampling distances that correspond to the differences between consecutive tap positions, is discussed in this section. The selection of steps σ_i and the calculation of repeated bits is performed as follows..

For a given set of tap positions $\mathcal{I}_0 = \{l_1, \ldots, l_n\}$, let $D = \{d_1, \ldots, d_{n-1}\}$ be the corresponding set of differences between the consecutive tap positions. The sequence of sampling distances σ_i between the observed outputs w^{t_i} and $w^{t_{i+1}}$ is defined as:

$$\begin{cases} \sigma_{1+p(n-1)} = d_1, \\ \sigma_{2+p(n-1)} = d_2, \\ \qquad \vdots \\ \sigma_{n-1+p(n-1)} = d_{n-1}, \end{cases} \tag{4.11}$$

for $p = 0, 1, 2, \ldots$. That is, the first $n-1$ sampling distances are taking values exactly from the set D so that $\sigma_1 = d_1, \sigma_2 = d_2, \ldots, \sigma_{n-1} = d_{n-1}$. Then, the next $n-1$ sampling distances are again $\sigma_n = d_1, \sigma_{n+1} = d_2, \ldots, \sigma_{2n-2} = d_{n-1}$, and so on. This mode of the $GFSGA^*$ we denote as $GFSGA^*_{(2)}$. For this mode, using Proposition 4.3.3, we are able to calculate a lower bound on the number of repeated equations for every sampling step. Recall that at some sampling instance t_j there are some repeated bit(s) if and only if $\sigma^{(j-i)} = \sum_{h=j-i}^{j} \sigma_h$, $i = 0, 1, \ldots, j-1$, is equal to some $\sum_{p=r}^{m} d_p = l_m - l_r$, for some $1 \le r \le m \le n-1$. In addition, if in the same column in the scheme of differences (which is equivalent to considering a fixed r and all the values $m \ge r$) we have more matchings $\sigma^{(j-i)} = \sum_{p=r}^{m} d_p$, then only one bit is counted.

Hence, taking the first $n-1$ sampling steps to be $\sigma_1 = d_1, \sigma_2 = d_2, \ldots, \sigma_{n-1} = d_{n-1}$, the scheme of differences (constructed for our set $D = \{d_1, \ldots, d_{n-1}\}$) and Proposition 4.3.3 imply that $q_1 \ge 1$, since at least $\sigma_1 = d_1$ is in Col. 1. Then $q_2 \ge 2$, since we have $\sigma^{(2-0)} = \sigma^{(2)} = \sigma_2$ is equal to d_2 in Col. 2., and $\sigma^{(2-1)} = \sigma^{(1)} = \sigma_1 + \sigma_2$ is equal to $d_1 + d_2$ in Col. 1. Continuing this process we obtain $q_3 \ge 3, \ldots, q_{n-1} \ge n-1$, which in total gives at least

$$1 + 2 + \ldots + (n-1) = \frac{n(n-1)}{2}$$

repeated equations for the first $n-1$ observed outputs. In the same way, at least $\frac{n(n-1)}{2}$ of bits are always repeated if further sampling at $n-1$ time instances is performed in accordance to (4.11). For instance, when $p = 1$ in (4.11) we have $\sigma_n = d_1$. In general, for $1 \le i \le n-1$ and $p \ge 0$ we have $q_{i+p(n-1)} \ge i$, where the sampling steps are defined by (4.11). We conclude this section with the following remarks.

Remark 4.3.4 *Both $GFSGA^*_{(1)}$ and $GFSGA^*_{(2)}$ depend less on the placement of the tap positions in comparison to GFSGA. Indeed, for both modes the sampling distances σ_i are selected with respect to a given placement of tap positions but regardless of what this placement in general might be. These modes are therefore more useful for cryptanalytic purposes rather than to be used in the design of an optimal allocation of tap positions (given the length of LFSR and the number of taps).*

Remark 4.3.5 *In the case of $GFSGA$ where we have equal distances between the observed outputs, one may notice that the sequence of numbers $r_i = \#\mathcal{I}_i$ is always an increasing sequence. On the other hand, the sequence of numbers $q_i = \#\mathcal{I}_i^*$ for $GFSGA^*$ may not be increasing at all. In connection to Open problem 2, neither $GFSGA^*_{(1)}$ nor $GFSGA^*_{(2)}$ automatically provides an optimal sequence of steps σ_i (which would imply the minimization of $T^*_{Comp.}$). This means that there exist cases in which any of the modes $GFSGA$, $GFSGA^*_{(1)}$ and $GFSGA^*_{(2)}$ may outperform the other two (cf. Table 4.8 and Table 4.9, Section 4.4.2).*

4.4 Comparision between $GFSGA$, $GFSGA^*_{(1)}$ and $GFSGA^*_{(2)}$

In this section we compare the performance of the three GFSGA modes when the tap positions are selected (sub)optimally by using the algorithms proposed in Section 4.2. Moreover, the case when the set of differences D forms a full positive difference set is also considered and compared to the algorithmic approach.

4.4.1 Overview of the algorithms for tap selection

As briefly mentioned in the introduction the concept of a full difference set, which ensures that all the entries in the set of all pairwise differences are different, is not a very useful criterion for tap selection. This is especially true when GFSGA-like cryptanalysis is considered as shown in Section 4.4.2. The same applies to the set of consecutive differences which may be taken to have mutually coprime entries which still does not ensure a sufficient cryptographic strength. Thus, there is a need for a more sophisticated algorithmic approach for designing (sub)optimally the placement of n tap positions for a given length L of the LFSR. The main idea behind the algorithms proposed in Section 4.2 is the use of the standard $GFSGA$ mode with a constant sampling rate for the purpose of finding (sub)optimal placement of tap positions.

The proposed algorithms for the selection of tap positions use the design rationales that maximize the resistance of the cipher to the standard mode of $GFSGA$. Instead of specifying the set $\mathcal{I}_0$, both methods aim at constructing the set D of consecutive differences which gives a low number of repeated equations (confirmed by computer simulations) for any constant sampling distance σ, which implies a good resistance to GFSGA-like methods. In both algorithms a quality measure for the choice of tap positions is the request that the optimal step of the $GFSGA$ attack is as small as possible. The selection of tap positions itself is governed by the general rule, which is achieving co-prime differences between the tap positions (**Step A**) together with distributing taps all over the register (thus maximizing $\sum_{i=1}^{n-1} d_i \leq L-1$, where $d_i = l_{i+1} - l_i$, $\mathcal{I}_0 = \{l_1, \ldots, l_n\}$ being a set of tap positions).

The first algorithm is designed to deal with situations when the size D is not large (say $\#D \leq 10$). In this case the algorithm performs an exhaustive search of all permutations of the set D (**Step B**) and gives as an output a permutation which ensures a maximal resistance to $GFSGA$. The complexity of this search is estimated as $O(n! \cdot K)$, where K corresponds to the complexity of calculation $T_{Comp.}$ for all

possible σ.

The main question which arises here is whether the performance of the mentioned algorithms can be improved by using some of the new presented modes in the previous sections. Unfortunately, the main reasons why $GFSGA$ can not be replaced by $GFSGA^*_{(1)}$ or $GFSGA^*_{(2)}$ are the following:

1. Apart from Remark 4.3.4, due to their low dependency on the choice of tap positions, neither $GFSGA^*_{(1)}$ nor $GFSGA^*_{(2)}$ mode (through Proposition 4.3.3) simply do not provide enough information that can be used to construct the tap positions with high resistance to GFSGA attacks in general. It is clear that the presented algorithms above only give a (sub)optimal placement of tap positions due to impossibility to test exhaustively all permutations of D and additionally to perform testing of all difference sets D is infeasible as well. An optimal placement of tap positions providing the maximum resistance to GFSGA attacks leads us back to Open problem 2.

2. One may notice that the main role of the constant step σ used in the design of the algorithm in Section 4.2 is to reduce the repetition of bits in general, since σ may take any value from 1 to L. This reduction of the repeated bits is significantly larger when using the constant step than any variable step of sampling in $GFSGA^*_{(1)}$ or $GFSGA^*_{(2)}$, due to their specific definitions given by (4.10) and (4.11).

Notice that some criteria for tap selection regarding the resistance to the inversion attacks were proposed in [42]. The difference between the first and last tap position should be near or equal to $L-1$, which turns out to be an equivalent criterion of maximization of the sum $\sum_{i=1}^{n-1} d_i \leq L-1$, as mentioned above. Generalized inversion attacks [43] performed on filter generators, with the difference between the first and last tap position equal to M $(= l_n - l_1)$, have the complexity approximately 2^M [43]. Hence, taking that $\sum_{i=1}^{n-1} d_i = L-1$, where $d_i = l_{i+1} - l_i$ (with tap positions $\mathcal{I}_0 = \{l_1, \ldots, l_n\}$), one of the criteria which thwarts (generalized) inversion attacks is easily satisfied.

In addition, one may also use a λth-order full positive difference set [42] for tap selection, that is, the set of tap positions $\mathcal{I}_0 = \{l_1, \ldots, l_n\}$ with as small as possible parameter $\lambda = \max_{1 \leq \sigma \leq M} |\mathcal{I}_0 \cap (\mathcal{I}_0 + \sigma)|$. If $\lambda = 1$, then $\mathcal{I}_0$ is a standard full positive difference set. As illustrated in Table 4.9, to provide a high resistance to GFSGA-like attacks, the set of tap positions may be a λth-order full positive difference set with higher values of λ. Note that in the case of inversion attacks, smaller λ is required. In other words, (λth-order) full positive difference sets do not provide the same resistance to inversion attacks and GFSGA-like attacks, when the selection of tap positions is considered.

4.4.2 Full positive difference sets versus algorithmic choice

In this section, we compare the performance of the three GFSGA modes by applying these attacks to a cipher whose tap positions are chosen using the algorithms given in Section 4.2 and in the case the tap positions form suitably chosen full positive

difference sets, respectively. We analyze the resistance to GFSGA attacks (using these two methods for tap selection) and conclude that full positive difference sets do not give an optimal placement of tap positions, i.e., they do not provide a maximal resistance to GFSGA-like cryptanalysis.

It is not difficult to see that the set of rules valid for our algorithms essentially require that we choose a set of tap positions $\mathcal{I}_0$ so that the corresponding set of consecutive differences D, apart from having different elements (possibly all), is also characterized by the property that these differences are coprime and in a specific order. The situation when taps are not chosen optimally, implying a high divisibility of the elements in D, is illustrated in Table 4.7. Denoting by $T_{Comp.}$, $T^*_{Comp.(1)}$ and $T^*_{Comp.(2)}$ the running time of the $GFSGA$, $GFSGA^*_{(1)}$ and $GFSGA^*_{(2)}$ mode, respectively, it is obvious that GFSGA is superior to other modes in most of the cases, as indicated in Table 4.7. In Table 4.8, we compare the performance of the

Table 4.7: Complexity comparision of all three GFSGA modes for "bad" tap choices.

L	(n,m)	D	$T_{Comp.}$	$T^*_{Comp.(1)}$	$T^*_{Comp.(2)}$
80	(9,2)	{12, 3, 6, 12, 6, 4, 24, 12}	$2^{43.97}$	$2^{67.97}$	$2^{62.97}$
120	(11,3)	{5, 10, 15, 4, 5, 10, 5, 15, 20, 25}	$2^{37.7}$	2^{63}	$2^{69.7}$
160	(15,6)	{14, 7, 3, 14, 7, 7, 14, 7, 14, 28, 7, 14, 14, 7}	$2^{32.97}$	$2^{32.97}$	$2^{50.97}$

three modes, if the tap positions (sets of differences D) are chosen suboptimally according to rules and algorithms given in Section 4.2. Thus, if the tap positions

Table 4.8: Complexity comparison of GFSGA modes - algorithmic selection of taps.

L	(n,m)	D	$T_{Comp.}$	$T^*_{Comp.(1)}$	$T^*_{Comp.(2)}$
80	(7,2)	{5, 13, 7, 26, 11, 17}	$2^{69.97}$	$2^{63.97}$	$2^{59.97}$
120	(13,3)	{5, 7, 3, 13, 6, 11, 5, 11, 7, 13, 21, 17}	$2^{99.7}$	2^{104}	$2^{78.7}$
160	(17,6)	{5,11,4,3,7,9,1,2,23,15,5, 13, 7, 26, 11, 17}	$2^{86.97}$	$2^{79.97}$	$2^{41.97}$
200	(21,7)	{3, 7, 9, 13, 18, 7, 9, 1, 2, 9, 1, 2, 23, 15, 5, 13, 7, 26, 11, 17}	$2^{108.9}$	$2^{96.93}$	$2^{68.93}$

are chosen according to the rules and algorithms given in Section 4.2, it turns out that $GFSGA^*_{(1)}$ and $GFSGA^*_{(2)}$ modes are more efficient than $GFSGA$.

In Table 4.9, we compare the resistance of a nonlinear filter generator (specified by L, n and m) to different GFSGA modes regarding the design rationales behind the choice of tap positions. Namely, for the same cipher (in terms of the parameters above), the attack complexities are evaluated for tap positions that form (suitable) full positive differences sets and, respectively, for the choices of tap positions given in Table 4.8 (with a slight modification adopted for different parameters L, n and m). In general, the algorithmic approach gives a higher resistance to GFSGA-like cryptanalysis.

Remark 4.4.1 *Table 4.9 also indicates that an algorithmic choice of tap positions may provide significantly better resistance against GFSGA-like attacks compared to full positive difference sets (for various parameters n, m and L).*

Table 4.9: Complexity comparision - full positive difference sets versus algorithmic choice.

L	(n,m)	Tap positions - Full positive difference sets	$T_{Comp.}$	$T^*_{Comp.(1)}$	$T^*_{Comp.(2)}$
80	(7,2)	{1, 3, 8, 14, 22, 23, 26}	$2^{35.97}$	$2^{37.97}$	$2^{57.97}$
120	(13,3)	{1, 3, 6, 26, 38, 44, 60, 71, 86, 90, 99, 100, 107}	$2^{86.72}$	$2^{90.72}$	$2^{95.72}$
160	(15,4)	{1, 5, 21, 31, 58, 60, 63, 77, 101, 112, 124, 137, 145, 146, 152}	$2^{96.97}$	$2^{105.97}$	$2^{116.97}$
200	(17,5)	{1, 6, 8, 18, 53, 57, 68, 81, 82, 101, 123, 139, 160, 166, 169, 192, 200}	$2^{113.93}$	$2^{123.93}$	$2^{132.93}$

L	(n,m)	Set of consecutive differences D -algorithmic choice	λ	$T_{Comp.}$	$T^*_{Comp.(1)}$	$T^*_{Comp.(2)}$
80	(7,2)	{5, 13, 7, 26, 11, 17}	1	$2^{69.97}$	$2^{63.97}$	$2^{59.97}$
120	(13,3)	{5, 7, 3, 13, 6, 11, 5, 11, 7, 13, 21, 17}	3	$2^{99.7}$	2^{104}	$2^{78.7}$
160	(15,4)	{5, 3, 7, 1, 9, 17, 15, 23, 5, 13, 7, 26, 11, 17}	3	$2^{114.97}$	$2^{124.97}$	$2^{101.97}$
200	(17,5)	{7, 13, 10, 13, 7, 1, 9, 17, 15, 23, 5, 13, 7, 26, 11, 17}	3	$2^{120.93}$	$2^{120.93}$	$2^{113.93}$

4.4.3 Further examples and comparisons

In this section we provide a few examples which illustrate the sampling procedure and specification of repeated bits for the $GFSGA^*_{(1)}$ and $GFSGA^*_{(2)}$ modes. In both cases we consider the set of consecutive differences $D = \{5, 13, 7, 26, 11, 17\}$ (most of the differences being prime numbers) which corresponds to the set of tap positions $\mathcal{I}_0 = \{1, 6, 19, 26, 52, 63, 80\}$. We first consider the $GFSGA^*_{(1)}$ mode.

Example 4.4.2 *Let the set of tap positions be given by* $\mathcal{I}_0 = s^{t_1} = \{1, 6, 19, 26, 52, 63, 80\}$, *where* $L = 80$ *and* $F : GF(2)^7 \to GF(2)^2$ *(*$n = 7$, $m = 2$*). The set* $\mathcal{I}_0$ *is chosen according to the algorithms in Section 4.2 and it is most likely an optimal choice of tap positions for the given parameters* L, n *and* m. *Recall that the variable sampling steps* σ_i *for* $GFSGA^*_{(1)}$ *are determined by the maximum function used in relation (4.10). In Table 4.10, using the relation (4.10) we identify the repeated state bits until the inequality* $nc^* > L + R^*$ *is satisfied for some* c^*. *The total number of repeated equations over all observed outputs is* $R^* = \sum_{k=1}^{c^*-1} q_k = 67$, *where the number of outputs is* $c^* = 22$. *Note that the first observed output* w^{t_1} *has the preimage space of full size, and thus there are no repeated bits. Since we chose* $s^{t_1} = \mathcal{I}_0$, *the positions of repeated bits at the corresponding tap positions can be found and calculated as follows:*

- *The step* $\sigma_1 = 5$ *gives the maximal intersection between* s^{t_1} *and* $s^{t_2} = \{s^{t_1} + 5\} = \{6, 11, 24, 31, 57, 68, 85\}$, *i.e., we have*

$$\max_{1 \le \sigma_1 \le 80} \#\{s^{t_1} \cap (s^{t_1} + \sigma_1)\} = \max_{1 \le \sigma_1 \le 80} \#\{s^{t_1} \cap s^{t_2}\} = \{6\},$$

  which yields $q_1 = 1$. *The size of the preimage space is* $|S_{w^{t_2}}| = 2^{n-m-q_1} = 2^4$.

- *Assuming the knowledge of* $x^{t_2} \in S_{w^{t_2}}$ *and* $x^{t_1} \in S_{w^{t_1}}$, *we search for an optimal shift* σ_2 *of* s^{t_2} *so that* $q_2 = \#\{\{s^{t_1} \cup s^{t_2}\} \cap \{s^{t_2} + \sigma_2\}\}$ *is maximized. Note that*

Table 4.10: Repeated bits attained by sampling steps σ_i defined by (4.10).

i	Sets $\mathcal{I}_i^*$ (where $(k+1) \leftrightarrow s_k$)	q_i	σ_i
1	{6}	1	5
2	{19,24}	2	13
3	{26, 31, 44}	3	7
4	{52, 57, 70, 77}	4	26
5	{63, 68, 81, 88, 114}	5	11
6	{80, 85, 98, 105, 131, 142}	6	17
7	{85, 103}	2	5
8	{114, 147}	2	11
9	{131, 164, 175}	3	17
10	{118, 136}	2	5
11	{125, 138}	2	2
12	{131, 136, 182}	3	11
13	{138, 143, 156}	3	7
14	{164, 169, 182, 189}	4	26
15	{175, 180, 193, 200, 226}	5	11
16	{192, 197, 210, 217, 243, 254}	6	17
17	{197, 215}	2	5
18	{199, 217}	2	2
19	{210, 215, 261}	3	11
20	{217, 222, 235}	3	7
21	{243, 248, 261, 268}	4	26

at this point, $\{s^{t_1} + \sigma_1\} = s^{t_2}$ *is fixed. This gives* $\sigma_2 = 13$ *and* $q_2 = 2$*. The set of repeated bits is*

$$\max_{1 \le \sigma_2 \le 80} \#\{\{s^{t_1} \cup s^{t_2}\} \cap \{s^{t_2} + \sigma_2\}\} = \{19, 24\},$$

since $s^{t_3} = \{s^{t_2} + \sigma_2\} = \{19, 24, 37, 44, 70, 81, 98\}$*. The preimage space has the cardinality* $|S_{w^{t_3}}| = 2^{n-m-q_2} = 2^3$*.*

In this way, we can completely determine the preimage spaces and the positions of the repeated bits. Since for $i \in \{5, 6, 15, 16\}$ *we have* $q_i \ge n - m = 5$*, then* $2^{n-m-q_i} = 1$ *(by convention). Once the other values* q_j *have been computed, for* $j \in \{1, 2, \ldots, 21\} \backslash \{5, 6, 15, 16\}$*, the attack complexity can be estimated as*

$$T^*_{Comp._{(1)}} = 2^{n-m} \times 2^{n-m-q_1} \times \ldots \times 2^{n-m-q_{21}} \times L^3 \approx 2^{63.97}.$$

In the case of GFSGA, an optimal choice of the sampling distance is any $\sigma \in \{1, 13, 37\}$*. Each of these sampling steps requires* $c = 16$ *observed outputs and gives* $R = 24$ *repeated equations, where the set of all repeated bits is given by*

$$\{r_1, r_2, \ldots, r_{15}\} = \{0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 3, 3, 4, 4, 4\}.$$

The values $r_i = 0$*,* $i = 1, 2, 3, 4$*, mean that the corresponding sets* $\mathcal{I}_i$ *are empty. The attack complexity of GFSGA is then estimated as* $T_{Comp.} \approx 2^{69.97}$*.*

Example 4.4.3 *Now, for the same function* F *and the set of tap positions* $\mathcal{I}_0 = s^{t_1}$ *(or the set of differences* $D = \{5, 13, 7, 26, 11, 17\}$*), we illustrate the* $GFSGA^*_{(2)}$ *mode. In Table 4.11, we show all repeated bits for the sampling steps* σ_i *of the* $GFSGA^*_{(2)}$ *mode, which are defined by relation (4.11). Recall that the steps* σ_i *in this case are defined so that every* $n - 1 = 6$ *outputs are at distances* $d_i \in D$*. By*

Table 4.11: Repeated bits attained by sampling steps σ_i defined by (4.11).

i	Sets $\mathcal{I}_i^*$ (where $(k+1) \leftrightarrow s_k$)	q_i	σ_i
1	{6}	1	5
2	{19,24}	2	13
3	{26, 31, 44}	3	7
4	{52, 57, 70, 77}	4	26
5	{63, 68, 81, 88, 114}	5	11
6	{80, 85, 98, 105, 131, 142}	6	17
7	{85, 103}	2	5
8	{98, 103}	2	13
9	{105, 110, 123}	3	7
10	{131, 136, 149, 156}	4	26
11	{142, 147, 160, 167, 193}	5	11
12	{159, 164, 177, 184, 210, 221}	6	17
13	{164, 182}	2	5
14	{177, 182}	2	13
15	{184, 189, 202}	3	7
16	{210, 215, 228, 235}	4	26
17	{221, 226, 239, 246, 272}	5	11
18	{238, 243, 256, 263, 289, 300}	6	17
19	{243, 261}	2	5
20	{256, 261}	2	13
21	{263, 268, 281}	3	7

formula (4.7), the complexity of $GFSGA^*_{(2)}$ *is estimated as* $T^*_{Comp.(2)} \approx 2^{59.97}$*, and thus this mode outperforms both* $GFSGA$ *and* $GFSGA^*_{(1)}$*. The total number of repeated equations in this case is given by* $R^* = 72$*, for* $c^* = 22$ *observed outputs. Notice that both modes* $GFSGA^*_{(1)}$ *and* $GFSGA^*_{(2)}$ *required in total* 22 *outputs to construct an overdefined system of linear equations* ($nc^* > L + R^*$).

4.5 Employing GFSGA in other settings

The main limitation of GFSGA-like attacks is their large complexity when applied to standard filtering generators that only output a single bit each time the cipher is clocked. In addition, this generic method cannot be applied in a straightforward manner in the cryptanalysis of ciphers that use NFSRs. In this section, we discuss the possibility of improving the efficiency and/or applicability of GFSGA with variable sampling step for the above mentioned scenarios. It will be demonstrated that GFSGA with variable sampling step may be employed in combination with other cryptanalytic methods to handle these situations as well.

4.5.1 GFSGA applied to single-output nonlinear filter generators

The time complexity of GFSGA with variable sampling step is given by (4.7), i.e.,

$$T^*_{Comp.} = 2^{n-m} \times 2^{n-m-q_1} \times \ldots \times 2^{n-m-q_{c^*-1}} \times L^3,$$

and clearly when $m = 1$ the complexity becomes quickly larger than the time complexity of exhaustive search (for some common choices of the design parameters n and L). Based on annihilators in fewer variables of a nonlinear filtering function $f(x_1, \ldots, x_n)$, Jiao *et al.* proposed another variant of FSGA in [57]. The core idea of

this attack is to reduce the size of the preimage space via annihilators in fewer variables $g_1(x_{j_1}, \ldots, x_{j_d})$ and $g_2(x_{j_1}, \ldots, x_{j_d})$ such that $f(x_1, \ldots, x_n)g_1(x_{j_1}, \ldots, x_{j_d}) = 0$ and $(f(x_1, \ldots, x_n) \oplus 1)g_2(x_{j_1}, \ldots, x_{j_d}) = 0$, where $\{j_1, \ldots, j_d\} \subset \{1, \ldots, n\}$. It was shown that the time complexity of this attack is given by

$$T^{\Delta}_{Comp.} = ||S_{g_1=0}||^{c_1} \times ||S_{g_2=0}||^{c_2} \times L^{\omega},$$

where $||S_{g_i=0}||$, for $i = 1, 2$, is the size of preimage space of the annihilator g_i (restricted to the variables $\{j_1, \ldots, j_d\}$), $c^* = \lceil \frac{L}{d} \rceil$ is the number of sampling steps, $c^* = c_1 + c_2$ and $\omega = \log_2 7 \approx 2.807$ is the exponent of Gaussian elimination. In [57], it was also shown that this variant of FSGA could be applied to single-output filter generators. For instance, letting $L = 87$ and using a nonlinear Boolean functions $f(x_1, \ldots, x_6)$ as in "Example 2" in [57], it was demonstrated that the time complexity of this attack is only about 2^{80} operations, whereas the time complexity of FSGA is about 2^{87} operations in [57].

In a similar manner, the same approach leads to a reduction of time complexity when GFSGA with variable sampling step is considered. For instance, let the set of tap positions be given by $\mathcal{I}_0 = s^{t_1} = \{1, 6, 19, 26, 52, 63\}$ corresponding to the inputs $\{x_1, x_2, x_3, x_4, x_5, x_6\}$, respectively. Using the filtering function $f : \mathbb{F}_2^6 \to \mathbb{F}_2$ of "Example 2" in [57], one can deduce that $||S_{g_1(x_2,x_4,x_6)=0}|| = ||S_{g_2(x_2,x_4,x_6)=0}|| = 5$. Actually, we can only consider the tap positions $\{6, 26, 63\}$ with full positive difference set $\{20, 37\}$. Moreover, let us use the variable sampling steps $\sigma_i = 20$ and $\sigma_{i+1} = 37$, alternately. In such a case, the preimage space of annihilator can be further reduced to $||S^*_{g_1(x_2,x_4,x_6)=0}|| = ||S^*_{g_2(x_2,x_4,x_6)=0}|| \approx 2.5$ by using the repeated bits. The time complexity of GFSGA with variable sampling steps is about $5 \times 2.5^{42} \times 87^{2.807} \approx 2^{76.32} < 2^{80}$ operations. In particular, the number of variable sampling points is 43 since at the first sampling point 3 linear relations are obtained and the remaining 42 sampling points give $2 \times 42 = 84$ linear relations, thus in total $3 + 84 = 87 = L$ linear equations are derived. It directly means that our GFSGA with variable sampling step outperforms the variant of FSGA in [57].

Due to the small sized parameters L and n the above example does not illustrate a full potential of using GFSGA in cryptanalysis of single-output filtering generators. Its purpose is rather to show that GFSGA and its variants can be efficiently combined with other cryptanalytic methods. The most promising approach seems to be an interaction of GFSGA with algebraic attacks using small degree annihilators of restrictions of the filtering function f. Indeed, the use of repeated bits not only reduces the preimage space it essentially also fixes a subset of input variables and therefore these restrictions of f may have annihilators of very low degree. This implies the existence of additional low degree equations in state bits which may be either used for checking the consistency of the linear system and after all (for sufficiently large number of fixed variables) these equations become linear. It is beyond the scope of this paper to investigate further the performance of this combined method but we believe that this kind of attack may become efficient against single-output filter generators with standard choice of the parameters L and n.

4.5.2 Applying GFSGA to NFSR-based ciphers

A current tendency in the design of stream ciphers, motivated by efficient hardware implementation, is the use of NFSRs in combination with (rather simple) nonlinear filtering function. For instance, this idea was employed in the design of the famous stream ciphers Trivium [11] and Grain family [1]. Apparently, none of the GFSGA variants can be applied for recovering the initial state of these ciphers but rather for deducing certain internal state of the cipher. In this scenario, the complexity of GFSGA is directly related to the complexity of solving an overdefined system of low degree equations rather than a system of linear equations. More precisely, assuming that the length of NFSR is L bits, the algebraic degree of its update function is r, and the filtering function $F : GF(2)^n \to GF(2)^m$, then the time complexity of GFSGA is given by

$$T^{**}_{Comp.} = 2^{n-m} \times 2^{n-m-q_1} \times \ldots \times 2^{n-m-q_{c^*-1}} \times D^{\omega}, \tag{4.12}$$

where $D = \sum_{i=0}^{e \times r} \binom{L}{i}$, $\omega = 2.807$ is the coefficient of Gaussian elimination, c^* is the number of sampling steps, and e is closely related to the parameters n, m, L, c^* and specified tap positions. The complexity being much larger than for LFSR-based ciphers, due to the term D^{ω}, makes GFSGA methods practically inefficient.

However, one may mount another mode of internal state recovery attack which employs the GFSGA sampling procedure, but without solving systems of equations at all. More precisely, this new type of internal state recovery attack also employs the sampling of outputs within a certain sampling window which then allows us to efficiently recover a certain portion of internal state bits from the reduced preimage spaces corresponding to the observed outputs. To describe the attack in due detail, let us denote by p the distance between the last entry of NFSR (where NFSR is updated) and the tap position closest to this registry cell. We assume that this distance satisfies the inequality $(p-1) \times n > L$, where n is the number of inputs (tap positions) of a filtering function $F : \mathbb{F}_2^n \to \mathbb{F}_m$. Note that this condition implies that either p or n are relatively large. In such a case, let us choose the constant sampling steps $\sigma_i = 1$, for $i = 1, \ldots, p-1$, and assume the adversary can directly recover, say R_p internal state bits (in total), at these $p-1$ sampling instances. The remaining $L - R_p$ internal state bits are still unknown and the adversary can exhaustively guess these bits to recover the whole internal state. The process of identifying the correct internal state is as follows. For each possible internal state candidate, a portion of L keystream bits $Z^t = (z_1, \ldots, z_L)$ at time instance t is determined using a given encryption algorithm. Then, comparing L keystream bits $Z^{*t} = (z_1^*, \ldots, z_L^*)$ at time instance t generated by the cipher (with unknown secret internal state), we can distinguish the correct and wrong internal states by directly checking if $Z^t = Z^{*t}$. In particular, if $Z^t = Z^{*t}$, then the guessed internal state would be the correct one, otherwise another internal state candidate is considered. Consequently, the time complexity of this internal state recovery attack, assuming that remaining $L - R_p$ bits are guessed, is given by

$$T^{**}_{Comp.} = 2^{n-m} \times 2^{n-m-q_1} \times \ldots \times 2^{n-m-q_{p-2}} \times 2^{L-R_p}. \tag{4.13}$$

The memory complexity of this attack is only $(p-1) \times n \times 2^{n-1} + L$ bits, which are used to save all the element of preimage spaces and L keystream bits.

The following example illustrates an application of this approach to an NFSR-based cipher that largely resembles the NFSR used in the Grain-128 cipher. In particular, the process of recovering R_p internal state bits is described more thoroughly.

Example 4.5.1 *Let $L = 128$, $n = 8, m = 1$. The update function of NFSR is defined below (a slightly modified variant of the NFSR used in Grain-128 [1] without cubic and quartic terms):*

$$\begin{aligned} b_{t+128} &= 1 \oplus b_t \oplus b_{t+26} \oplus b_{t+56} \oplus b_{t+91} \oplus b_{t+96} \oplus b_{t+3}b_{t+67} \oplus b_{t+11}b_{t+13} \\ &\quad \oplus b_{t+17}b_{t+18} \oplus b_{t+27}b_{t+59} \oplus b_{t+40}b_{t+48} \oplus b_{t+61}b_{t+65} \oplus b_{t+68}b_{t+84}. \end{aligned}$$

The set of tap positions which we consider is given by $\mathcal{I}_0 = s^{t_1} = \{l_1, \ldots, l_8\} = \{1, 7, 21, 26, 52, 67, 89, 105\}$, and it corresponds to a full positive difference set $\{6, 14, 5, 26, 15, 22, 16\}$. The distance between the last tap position and the NFSR update position is $p = 128 - 105 = 23$. This means that if we consider an updated internal state bit (the first nonlinear bit) and constant sampling rate $\sigma_i = 1$, this bit will appear at the tap position after 23 sampling instances. At the same time, employing the fact that many of these bits appear at some tap positions (thus using the idea of GFSGA), the adversary directly obtains many bits corresponding to 128-bit internal state as follows:

1. *By relation (4.5) and sampling steps described in Section 4.3, collecting all repeated bits over $p-1$ observed outputs (which are on consecutive distances $\sigma_i = 1$), we determine all preimage spaces $S_{w^{t_i}}$ $(i = 1, \ldots, p-1)$ and their sizes.*

2. *Our approach implies that at the sampling instance i we recover (essentially guess) $n - q_i$ internal state bits which must match to one of the 2^{n-q_i-1} preimages. It is important to note that the bits which come from preimage spaces are the only candidates to be an internal state of the registry, since they are precisely determined by consecutive repetitions over $p-1$ observed outputs.*

Table 4.12 specifies the number of recovered internal state bits, and the sizes of corresponding preimage spaces. Denoting by R_p the total number of recovered bits, we can see (from Table 4.12) that $R_p = 8+8\times4+7+6\times8+5+4+3\times6 = 122 < 128$, where these bits are calculated using (4.5), for $\sigma_i = 1$, $(i = 1, \cdots, 22)$.

The adversary can further guess the remaining $L - R_p = 128 - 122 = 6$ internal state bits, and thus the time complexity, using (4.13), of this attack is about

$$T^{**}_{Comp.} = 2^{7+7\times4+6+5\times8+4+3+2\times6} \times 2^6 = 2^{106} < 2^{128}.$$

The data complexity of this attack is only about $22 + 128 = 150$ keystream bits. The memory complexity is upper bounded by $22 \times 8 \times 2^7 + 128 < 2^{15}$ bits, which corresponds to storing at most 2^7 elements from preimage spaces and 128 keystream bits. The success rate is close to one since there are $2^{7+7\times4+6+5\times8+4+3+2\times6+6} = 2^{106}$ internal state candidates in total and therefore only a small portion of about $2^{106} \times 2^{-128} = 2^{-22} < 1$ wrong internal state candidates can pass the test.

Table 4.12: Recovered bits obtained by sampling step $\sigma_i = 1$

i	Recovered bits of internal state	q_i	The size of preimage space
1	8	0	2^7
2	8	0	2^7
3	8	0	2^7
4	8	0	2^7
5	7	1	2^6
6	6	2	2^5
7	6	2	2^5
8	6	2	2^5
9	6	2	2^5
10	6	2	2^5
11	6	2	2^5
12	6	2	2^5
13	6	2	2^5
14	5	3	2^4
15	4	4	2^3
16	3	5	2^2
17	3	5	2^2
18	3	5	2^2
19	3	5	2^2
20	3	5	2^2
21	3	5	2^2

Example 4.5.1 demonstrates that GFSGA-like attacks can be applied to NFSR-based stream ciphers without employing any structural properties of the filtering function. The following example illustrates an application of GFSGA to a hybrid NFSR/LFSR-based cipher whose design is very similar to Grain-128 cipher. The major difference is the key length, since our variant assumes that the key length is $L = 256$ bits rather than 128-bit key used in Grain-128 [1].

Example 4.5.2 *Let $L = 256$, $n = 17, m = 1$. The internal state of our variant of Grain-128 consists of one 128-bit LFSR and one 128-bit NFSR, whose state bits are denoted by $(s_0, \cdots, s_{127})$ and $(b_0, \cdots, b_{127})$, respectively. Their update functions are defined respectively as follows (see also [1]):*

$$\begin{aligned} s_{t+128} &= s_t \oplus s_{t+7} \oplus s_{t+38} \oplus s_{t+70} \oplus s_{t+81} \oplus s_{t+96} \\ b_{t+128} &= s_t \oplus b_t \oplus b_{t+26} \oplus b_{t+56} \oplus b_{t+91} \oplus b_{t+96} \oplus b_{t+3}b_{t+67} \oplus b_{t+11}b_{t+13} \\ &\quad \oplus b_{t+17}b_{t+18} \oplus b_{t+27}b_{t+59} \oplus b_{t+40}b_{t+48} \oplus b_{t+61}b_{t+65} \oplus b_{t+68}b_{t+84} \end{aligned} \tag{4.14}$$

For this variant of Grain-128 cipher we consider the same set of tap positions that are used in the standard Grain-128 cipher, i.e., the tap position are $A = \{2, 12, 15, 36, 45, 64, 73, 89, 95\}$ for the NFSR and $B = \{8, 13, 20, 42, 60, 79, 93, 95\}$ for the LFSR. Note that the largest tap index in A is 95, and the NFSR is updated at position 128, i.e., their distance is $p = 128 - 95 = 33$. Similarly as in Example 4.5.1, sampling at the constant rate $\sigma_i = 1$, Table 4.13 specifies the number of recovered (repeated) bits of internal state, and the size of preimage spaces. Thus, the adversary can directly obtain $17 + 227 = 244 < 256$ internal state bits. The remaining $L - R_p = 256 - 244 = 12$ internal state bits can then be guessed, which then leads to a recovery of the whole 256-bit internal state. Therefore, the time complexity of this attack is about

$$T^{**}_{Comp.} = 2^{16+196} \times 2^{12} = 2^{224} < 2^{256}.$$

Table 4.13: Repeated bits attained by sampling step $\sigma_i = 1$

i	Recovered bits of internal state	q_i	The size of preimage space
1	17	0	2^{16}
2	16	1	2^{15}
3	15	2	2^{14}
4	15	2	2^{14}
5	14	3	2^{13}
6	13	4	2^{12}
7	12	5	2^{11}
8	12	5	2^{11}
9	10	7	2^{9}
10	9	8	2^{8}
11	9	8	2^{8}
12	9	8	2^{8}
13	9	8	2^{8}
14	8	9	2^{7}
15	8	9	2^{7}
16	7	10	2^{6}
17	7	10	2^{6}
18	6	11	2^{5}
19	4	13	2^{3}
20	4	13	2^{3}
21	3	14	2^{2}
22	2	15	2
23	2	15	2
24	2	15	2
25	2	15	2
26	2	15	2
27	2	15	2
28	2	15	2
29	2	15	2
30	2	15	2
31	2	15	2
	$\sum = 227$		$\prod = 2^{196}$

The above example demonstrates that GFSGA-like cryptanalysis is also applicable to hybrid NFSR/LFSR-based ciphers. In particular, it is shown that the tap positions have a very important impact on the security of NFSR/LFSR-based ciphers.

Remark 4.5.3 *In difference to the time-memory-data trade-off attacks or algebraic attacks, this attack has more favorable data and memory complexity. For instance, in Example 4.5.2, the data complexity of this attack is only about* $32+229 = 261 \approx 2^8$ *keystream bits. Namely, in the first step we use 32 sampling instances to determine all specified preimage spaces and their sizes under constant sampling rate* $\sigma_i = 1$*, and in the second step we need to use about* 229 *fresh keystream bits to determine the correct state. Notice that the memory complexity of this attack is only about* $32 \times 17 \times 2^{16} + 256 \approx 2^{25}$ *bits. On the other hand, if the filtering function is* $f : GF(2)^{17} \to GF(2)^m, m > 4$*, then the time complexity of this attack is less than* 2^{128} *operations. It implies that this attack would outperform the time-memory-data trade-off attack for* $m > 4$*.*

4.5.3 Grain-128 tap selection

We have already remarked that the tap selection for both SOBER-t32 and SFINKS was not optimal with respect to their resistance to GFSGA cryptanalysis (Section

4.2). We show that the same is true when Grain-128 is considered, thus there exist better selections that ensure greater resistance to GFSGA-like cryptanalysis.

We assume that either LFSR or NFSR, whose tap positions are given in Example 4.5.2, of Grain-128 are used as state registers in a filter generator and we apply different modes of GFSGA to these schemes. In the case when the LFSR of Grain-128 is employed in such a scenario then the complexities of the three different modes of GFSGA are given as,

Table 4.14: Time complexity of different modes of GFSGA on LFSR of Grain-128

$T_{Comp.}$	$T^*_{Comp._{(1)}}$	$T^*_{Comp._{(2)}}$
2^{108}	2^{125}	2^{118}

Using our algorithm for finding a (sub)optimal placement of tap positions, instead of using the set $A = \{2, 12, 15, 36, 45, 64, 73, 89, 95\}$, we find another set of tap positions given as $\{1, 16, 27, 54, 71, 95, 108, 127\}$ which gives the following complexities,

$$T_{Comp.} = 2^{129}, \quad T^*_{Comp._{(1)}} = 2^{132}, \quad T^*_{Comp._{(2)}} = 2^{123}.$$

A similar improvement can also be achieved when the tap positions of NFSR in Grain-128 are considered. In this case the original placement of tap positions (the set B in Example 4.5.2) gives the following complexities,

Table 4.15: Time complexity of different modes of GFSGA on NFSR of Grain-128

$T_{Comp.}$	$T^*_{Comp._{(1)}}$	$T^*_{Comp._{(2)}}$
2^{114}	2^{125}	2^{122}

On the other hand, our algorithm suggest somewhat better allocation of these taps given by $\{3, 10, 29, 42, 59, 67, 88, 103, 126\}$, which then induces the following complexities of the three GFSGA modes,

$$T_{Comp.} = 2^{130}, \quad T^*_{Comp._{(1)}} = 2^{139}, \quad T^*_{Comp._{(2)}} = 2^{125}.$$

Chapter 5

Estimating the algebraic properties of Boolean functions for large n

The security of these LFSR-based stream ciphers heavily relies on the algebraic properties of the used Boolean function. Over the last decades, Boolean functions satisfying some particular cryptographic properties (such as high nonlinearity, high algebraic immunity (AI) etc.) have been studied [13, 38, 127, 124]. The concept of algebraic immunity for an arbitrary Boolean function f was introduced in [79] and it reflects the resistance of a Boolean function f against AA. More precisely, this criterion measures the minimum algebraic degree of its annihilators, i.e., $AI_f = \min_{\deg(g)}\{A(f), A(f \oplus 1)\}$, where $A(f) = \{g : fg = 0, g \neq 0\}$ and $A(f \oplus 1) = \{g : (f \oplus 1)g = 0, g \neq 0\}$. It was shown that an optimal resistance of a Boolean function f against AA is achieved if $AI_f = \lceil n/2 \rceil$. On the other hand, a Boolean function with an optimal AI still cannot adequately ensure a good resistance against FAA that use the existence of the function pairs (g, h) (with algebraic degree $\deg(g)$ and $\deg(h)$ respectively) such that $fg = h$ and $\deg(g) + \deg(f)$ is not large [27, 81]. The value of $\deg(g) + \deg(h)$ measures the resistance of a Boolean function against FAA. An optimal resistance of Boolean functions (used in LFSR-based stream ciphers) against FAA implies that the minimum values of $\deg(g) + \deg(h)$ is always equal to n for any function pairs (g, h) such that $fg = h$, though such functions are very rare. In addition, it was shown that for balanced Boolean functions $\deg(g) + \deg(h) \geq n$ if and only if either $n = 2^k$ or $n = 2^k + 1$ for some positive integer k [67].

As mentioned in Chapter 1, the first algorithm for determining the existence of annihilators of degree d of a Boolean function with n variables, with time complexity about $O(D^3)$ ($D = \sum_{i=0}^{d} \binom{n}{i}$), was proposed in [25]. At FSE 2006, an algorithm for checking the existence of annihilators or multiples of degree less than or equal to d was introduced in [30] with time complexity of about $O(n^d)$ operations for an n-variable Boolean function. At EUROCRYPT 2006, based on the multivariate polynomial interpolation, Armknecht *et al.* [3] proposed an algorithm for computing $AI = d$ of a Boolean function with n variables [3] requiring $O(D^2)$ operations, where

$D = \sum_{i=0}^{d} \binom{n}{i}$. Moreover, an algorithm for determining the immunity against FAA was also presented running in time complexity of about $O(D^2E)$ operations for an n-variable Boolean function, where $E = \sum_{i=0}^{e} \binom{n}{i}$ and d is generally much smaller than e, $(\deg(g), \deg(h)) = (d, e)$. At ACISP 2006, an algorithm to evaluate the resistance of Boolean functions against FAA was developed in [8], whose time complexity is about $O(DE^2+D^2)$ operations for an n-variable Boolean function. At INDOCRYPT 2006, based on the Wiedemann's algorithm, Didier proposed a new algorithms to evaluate the resistance of an n-variable Boolean functions against AA and FAA in [29] with time complexity of about $O(n^{2n}D)$ operations and a memory complexity of about $O(n^{2n})$. Finally, Jiao *et al.* [56] revised the algorithm of [3] to compute the resistance against AA and FAA, reducing the complexity to $O(D^{2\pm\varepsilon})$ operations, where $\varepsilon \approx 0.5$ and D is the same as above.

The purpose of this chapter is to present an efficient probabilistic algorithm for determining the resistance of a random Boolean function against AA and FAA. A suitable choice of input parameters gives a high success rate of the algorithm so that the estimates are correct with probability very close to one. The algorithm employs partial linear relations, derived form the decomposition of an arbitrary nonlinear Boolean function into many small partial linear subfunctions by using the disjoint sets of input variables. A general probabilistic decomposition algorithm for nonlinear Boolean functions is given along with the sufficient conditions regarding the existence of low degree annihilators (or multipliers). This probabilistic algorithm provides a new framework for estimating the resistance of Boolean function against AA and FAA requiring only about $O(n^2 2^n)$ operations (for an n-variable Boolean function), thus offering much less complexity at the price of being probabilistic. The lower and upper bound on AI and FAA that we derive appears to be very tight for randomly selected Boolean functions thus giving a close estimate of the algebraic properties for large n where due to computational complexity the deterministic algorithms cannot be applied. Several examples are provided justifying the tightness of our bounds when compared to the actual algebraic properties of a given function for relatively small values of n for which the deterministic algorithms could be applied.

Results of this chapter are published in [120] and it is organized as follows. In Section 5.1, a new concept of partial linear relations decomposition is introduced, and then a general dissection algorithm for nonlinear Boolean functions is proposed. An efficient algorithm for determining the resistance of Boolean functions (with relatively large input variables n) against AA and FAA is descried in Section 5.2.

5.1 A probabilistic decomposition algorithm for nonlinear Boolean functions

In this section, a probabilistic decomposition algorithm for nonlinear Boolean functions which decomposes any Boolean functions into a set of partial linear relations is discussed. This decomposition is generic, deterministic and valid for arbitrary Boolean functions (fully specifying a given function) but it is not unique. The consequence is that different choices of such a decomposition may yield different estimates of algebraic properties, though since the number of these decompositions is not large

the algorithm may exhaustively check for the best decomposition. For brevity, in the result below we use the notation introduced by the following definition.

Definition 5.1.1 *Let $f \in \mathcal{B}_n$ be a nonlinear Boolean function, and $X = (x_1, \ldots, x_n) \in GF(2)^n$, $X'_i = (x_{j_1}, \ldots, x_{j_i}) \in GF(2)^i$, $X''_{n-i} = (x_{j_{i+1}}, \ldots, x_{j_n}) \in GF(2)^{n-i}$, where $\{j_1, \ldots, j_i\} \subset \{1, \ldots, n\}$, $\{j_{i+1}, \ldots, j_n\} \subset \{1, \ldots, n\}$ and $\{j_1, \ldots, j_i\} \cap \{j_{i+1}, \ldots, j_n\} = \emptyset$. If by fixing $X'_i = a$, the function $f(a, X''_{n-i}) = f_{X'_i=a}(X''_{n-i})$ is an $(n-i)$-variable linear subfunction or a constant function, then $f_{X'_i=a}(X''_{n-i})$ is called a partial linear relation with respect to $a \in GF(2)^i$. The set of all partial linear relations with $n-i$ variables is denoted by $\mathbb{L}_{n-i}$.*

Theorem 5.1.2 *Let $X = (x_1, \ldots, x_n) \in GF(2)^n$, and $D_i \subseteq \{L(X''_{n-i}) \mid L(X''_{n-i}) = c \cdot X''_{n-i} \oplus b, c \in GF(2)^{n-i}, b \in GF(2)\}$. Then given any nonlinear Boolean function $f \in \mathcal{B}_n$, there exist $B_i \subseteq GF(2)^i$ and $B'_i = B_i \times GF(2)^{n-i}$ such that $\bigcup_{i=1}^{n-1} B'_i = GF(2)^n$ and $B'_{i_1} \cap B'_{i_2} = \emptyset$, $(1 \leq i \leq n-1,\ 1 \leq i_1 < i_2 \leq n-1)$ so that f can be decomposed and represented as below:*

$$f(X) = f(X'_i, X''_{n-i}) = \sum_{i=1}^{n-1} \sum_{\sigma^{(i)} = (\sigma^{(i)}_{j_1}, \ldots \sigma^{(i)}_{j_i}) \in B_i} \left(\prod_{s=j_1}^{j_i} (x_s \oplus \sigma^{(i)}_s \oplus 1) \right) \cdot f(\sigma^{(i)}, X''_{n-i}) \quad (5.1)$$

where for any $\sigma^{(i)} \in B_i$ we have $f(\sigma^{(i)}, X''_{n-i}) \in D_i$.

Proof: For any nonlinear Boolean function $f \in \mathcal{B}_n$, for a given $X'_i = (x_{j_1}, \ldots x_{j_i}) = \sigma^{(i)} \in GF(2)^i$, the restriction $f(\sigma^{(i)}, X''_{n-i}) = f_{X'_i=\sigma^{(i)}}(X''_{n-i})$ is either a partial linear relation or a nonlinear function. Let $B_i = \{X'_i = \sigma^{(i)} \mid f(\sigma^{(i)}, X''_{n-i}) \in D_i, \sigma^{(i)} \in GF(2)^i\}$ be a collection of those $\sigma^{(i)} \in GF(2)^i$ for which $f(\sigma^{(i)}, X''_{n-i})$ is linear (affine) function in X''_{n-i} variables, where B_i may be empty. Moreover, if some fixed $X'_i \in GF(2)^i \setminus B_i$, let $X'_{i+1} = (X'_i, x_{j_{i+1}})$, then either $X'_{i+1} \in B_{i+1}$ or not. If $X'_{i+1} \in GF(2)^{i+1} \setminus B_{i+1}$, then we can increase the size of X'_{i+1} to X'_{i+2}. Iteratively, we reach the case $i = n-1$ for which $X'_{n-1} \in B_{n-1}$ always holds. Consequently, we can obtain a collection $B_i \subseteq GF(2)^i$ and $B'_i = B_i \times GF(2)^{n-i}$ such that $\bigcup_{i=1}^{n-1} B'_i = GF(2)^n$ and $B'_{i_1} \cap B'_{i_2} = \emptyset$, $1 \leq i_1 < i_2 \leq n-1$, where $1 \leq i \leq n-1$ and for any $\sigma^{(i)} \in B_i$ we have $f(\sigma^{(i)}, X''_{n-i}) \in D_i$. ∎

The following corollary is an easy consequence of the above result.

Corollary 5.1.3 *Using the notation of Theorem 5.1.2 the sets $B_i, (i = 1, \ldots, n-1)$ satisfies the relations below.*

1. $\sum_{i=1}^{n-1} ||B_i|| \times 2^{n-i} = 2^n$, $||B_i|| \leq 2^i$.
2. $||B_i|| \leq ||B_j||$ *for non-empty sets B_i and B_j employed in decomposition (5.1),* $(i < j)$.
3. *If $f(x_1, \ldots, x_n)$ is an affine function, then $||B_1|| = 2$ and $||B_i|| = 0, (i = 2, \ldots, n-1)$.*

4. *If* $f(x_1,\ldots,x_n) = x_1 \cdot x_2 \ldots x_n$, *then* $||B_{n-1}|| = 2$ *and* $||B_i|| = 1, (i = 1,\ldots,n-2)$.

5. *If* $f(x_1,\ldots,x_n)$ *is a full term function, then* $||B_{n-1}|| = 2^{n-1}$ *and* $||B_i|| = 0, (i = 1,\ldots,n-2)$.

Example 5.1.4 *Let* $f(x_1,\ldots,x_4) = x_1 \oplus x_4 \oplus x_1x_2 \oplus x_1x_2x_3$. *To write the function* f *in the form (5.1), we need to fix particular coordinates, so that the restrictions of the function* f *are linear or constant. For instance, by fixing* $x_1 = 0$, *or* $x_2 = 0$, *or* $x_2 = 1$ *and* $x_3 = 0$, *or* $x_2 = 1$ *and* $x_3 = 1$, $f(x_1,\ldots,x_4)$ *will be decomposed into linear functions. More precisely (neglecting the other cases):*

1. *If* $x_2 = 0$, *then* $f(x_1, 0, x_3, x_4) = x_1 \oplus x_4$. *The corresponding set of fixed coordinates (a single coordinate in this case) is* $B_1^{(x_2)} = \{0\}$, *and the corresponding set of linear functions is* $D_1 = \{x_1 \oplus x_4\}$.

2. *If* $(x_2, x_3) = (1, 0)$, *then* $f(x_1, 1, 0, x_4) = x_4$, *and if* $(x_2, x_3) = (1, 1)$ *then* $f(x_1, 1, 1, x_4) = x_1 \oplus x_4$. *The corresponding set of fixed coordinates (2-tuples) is* $B_2^{(x_2,x_3)} = \{(1, 0), (1, 1)\}$. *The corresponding set of linear relations is* $D_2 = \{x_4, x_1 \oplus x_4\}$.

We have that $B_3 = \emptyset$. *Clearly,*

$$||B_1|| \times 2^3 + ||B_2|| \times 2^2 + ||B_3|| \times 0 = 1 \times 2^3 + 2 \times 2^2 + 0 = 2^4,$$

which means that the union of subsets (subspaces) of $GF(2)^4$ *with fixed coordinates* $x_2 = 0$, $(x_2, x_3) = (1, 0)$ *and* $(x_2, x_3) = (1, 1)$ *actually gives the whole space* $GF(2)^4$, *i.e.,*

$$\{(x_1, 0, x_3, x_4) \mid x_i \in GF(2),\ i = 1, 3, 4.\} \cup \{(x_1, 1, 0, x_4) \mid x_i \in GF(2),\ i = 1, 4.\}$$

$$\cup\{(x_1, 1, 1, x_4) \mid x_1, x_4 \in GF(2)\} = GF(2)^4.$$

Then the function f *can be written as:*

$$\begin{aligned} f(x_1,x_2,x_3,x_4) &= \sum_{\sigma=(x_2)\in B_1} \left(\prod_{s=2}^{2}(x_s \oplus \sigma_s \oplus 1)\right) \cdot f_{x_2=\sigma^{(1)}}(x_1,x_3,x_4) \\ &\oplus \sum_{\sigma=(x_2,x_3)\in B_2} \left(\prod_{s=2}^{3}(x_s \oplus \sigma_s \oplus 1)\right) \cdot f_{(x_2,x_3)=\sigma^{(2)}}(x_1,x_4) \\ &= (x_2 \oplus 1)(x_1 \oplus x_4) \oplus x_2(x_3 \oplus 1)x_4 \oplus x_2x_3(x_1 \oplus x_4). \end{aligned}$$

The above result immediately leads to the following algorithm which decomposes an arbitrary Boolean function into a set of partial linear relations. We notice that the output of the algorithm heavily depends on the given choice (order) of variables which are to be fixed during its execution, see also Remark 5.1.5. In other words, the decomposition into linear subfunctions with respect to the cardinalities of B_i is quite likely not optimal and therefore a more refined search for the best decomposition (out of $n!$ possible ones) is later proposed, namely Algorithm 2.

Algorithm 1 (Partial Linear Relations Decomposition)

Step 1 For a given n-variable Boolean function $f \in \mathcal{B}_n$, let $k = \lceil \log_2 n \rceil$. Set counters $T_{B_i} = 0, (i = 1, \ldots, n-1)$. Without loss of generality, we assume the fixed decomposition order of $(n-1)$ variables to be $(x_1 \to x_2 \to \ldots \to x_{n-1})$.

Step 2 For each $x_1 = a_1 \in GF(2)$, randomly choose different 2^k pairs $(\alpha_{n-1}^j, \beta_{n-1}^j)$, $\alpha_{n-1}^j, \beta_{n-1}^j \in GF(2)^{n-1}$, for $j = 1, \ldots, 2^k$. Let $g_1(x_2, \ldots, x_n) = f_{x_1=a_1}(x_2, \ldots, x_n)$. For each pair $(\alpha_{n-1}^j, \beta_{n-1}^j)$, perform the linear relation test below :

$$g_1(\alpha_{n-1}^j \oplus \beta_{n-1}^j) = g_1(\alpha_{n-1}^j) \oplus g_1(\beta_{n-1}^j) \oplus g_1(0, \ldots, 0).$$

(2.1) If all 2^k pairs $(\alpha_{n-1}^j, \beta_{n-1}^j)$ pass this linear test (thus satisfy the above equality), then let $T_{B_1} = T_{B_1} + 1$.

(2.2) Otherwise, for each $(x_1, x_2) = a_2 \in GF(2)^2$, randomly choose different 2^k pairs $(\alpha_{n-2}^j, \beta_{n-2}^j)$, $\alpha_{n-2}^j, \beta_{n-2}^j \in GF(2)^{n-2}$, for $j = 1, \ldots, 2^k$. Let $g_2(x_3, \ldots, x_n) = f_{(x_1,x_2)=a_2}(x_3, \ldots, x_n)$ and again for each pair $(\alpha_{n-2}^j, \beta_{n-2}^j)$ perform 2^k linear relation tests:

$$g_2(\alpha_{n-2}^j \oplus \beta_{n-2}^j) = g_2(\alpha_{n-2}^i) \oplus g_2(\beta_{n-2}^i) \oplus g_2(0, \ldots, 0), \quad j = 1, \ldots, 2^k.$$

(2.2.1) If all 2^k pairs $(\alpha_{n-2}^j, \beta_{n-2}^j)$ pass the linear relation test, then let $T_{B_2} = T_{B_2} + 1$.

Otherwise, repeat the above steps by increasing the size of input variables, thus increase $i \to i+1$ and use $(x_1, \ldots, x_{i+1}) = a_{i+1} \in GF(2)^{i+1}$, for $i \le n-k$. For any such a_{i+1} perform 2^k linear tests for randomly chosen pairs $(\alpha_{n-i-1}^j, \beta_{n-i-1}^j) \in GF(2)^{n-i-1} \times GF(2)^{n-i-1}$, and update the values $T_{B_{i+1}}$.

For $i = n-k+1, \ldots, n-2$, randomly choose different 2^{n-i} pairs $(\alpha_{n-i}^j, \beta_{n-i}^j) \in GF(2)^{n-i} \times GF(2)^{n-i}$, for $j = 1, \ldots, 2^{n-i}$, and check whether all 2^{n-i} pairs can pass the linear relation test or not using $g_i = f_{(x_1,\ldots,x_i)=a_i}(x_{i+1}, \ldots, x_n)$.

Step 3 Return the values of $||B_i|| = T_{B_i}$, for $i = 1, \ldots, n-1$.

To estimate the success rate of this algorithm, we notice that each $g_i = f_{(x_1,\ldots,x_i)=a_i}($ $x_{i+1}, \ldots, x_n)$ (for different $a_i \in GF(2)^i$) can pass all 2^k linear relation tests only with a probability $\frac{1}{2^{2^k}}$ using 2^k random pairs, for $i = 1, \ldots, n-k$. However, there are $\sum_{i=1}^{n-k} ||B_i||$ subfunctions which need to be checked. This implies that there are about

$$\sum_{i=1}^{n-k} ||B_i|| \times 2^{-2^k} \le 2^{n-1} \times 2^{-n} = \frac{1}{2} < 1$$

nonlinear subfunctions g_i that could pass the linear relation tests.

Moreover, when $i \in [n-k+1, n-2]$, then each $g_i = f_{(x_1,\ldots,x_i)=a_i}(x_{i+1}, \ldots, x_n)$ only has 2^{n-i} input values in total. In this case, if g_i is a nonlinear function, the

probability of passing the linear relation tests is only $\frac{1}{2^{2^{n-i}}}$, using 2^{n-i} random pairs. For instance, if $n-i=3$, the probability is about $\frac{1}{2^8} \approx 0.0039$. But for $n-i=2$, to further improve the accuracy of the linear relation tests in practice, we can slightly increase the numbers of testing pairs to 6. In fact, if $n-i=2$, there are $2^2=4$ different input values for each g_{n-2} in total, which gives $\binom{4}{2}=6$ different pairs, i.e., $\{(11,00),(11,01),(11,10),(00,01),(00,10),(01,10)\}$. Obviously the probability of passing the six linear relation tests is 0, for any 2-variable nonlinear Boolean function g_{n-2}. Therefore, the success rate of this algorithm is about $p=1$.

On the other hand, the time complexity of this algorithm is dominated by Step 2, i.e.,

$$T_{complexity} = \sum_{i=1}^{n-k} 2^i \times 2^k + \sum_{j=n-k+1}^{n-2} 2^j \times 2^{n-j}.$$

Moreover, we have

$$\begin{aligned} T_{complexity} &= 2^k \sum_{i=1}^{n-k} 2^i + \sum_{j=n-k+1}^{n-2} 2^j \times 2^{n-j} \\ &= 2^{n+1} - 2^{k+1} + 2^n \times (n-2-(n-k+1)+1) \\ &= k \times 2^n - 2^{k+1} \\ &< k \times 2^n. \end{aligned}$$

where $k=\log_2 n$. Therefore, the time complexity of this algorithm is about $(\log_2 n)\times 2^n$ operations. The memory complexity is only about $O(2n)$ n-bit, which is mainly used to save the parameters T_{B_i} and the vectors in B_i that define the decomposition of $GF(2)^n$.

Remark 5.1.5 *In Step 1, for different orders of $(n-1)$-variable, this algorithm will return different values of $||B_i||$, for $i=1,\ldots,n-1$. It is clear that there are $\binom{n}{n-1} \times (n-1)! = n!$ ordered choices for a given n-variable function f. Therefore, there are $n!$ different values for $||B_i||$. However, it is computationally infeasible to calculate all these values if n is relatively large. We also notice that the approach taken in [30], which employs small subfunctions of f, allows that these subfunctions are also nonlinear. Our algorithm, due to strict linear decomposition, does not allow the use of nonlinear subfunctions.*

Algorithm 1 essentially provides an upper bound (for a fixed decomposition order) on the algebraic degree of annihilators of f due to the following result.

Theorem 5.1.6 *With the same notation used in Theorem 5.1.2, if Boolean function $f \in \mathcal{B}_n$ can be decomposed (with $B_i, i=(1,\ldots,n-1)$) by using Algorithm 1, then there is at least an annihilator $g \in \mathcal{B}_n$ with $deg(g) \leq \lambda+1$ such that $f \cdot g = 0$, where $\lambda = \min\{i \mid ||B_i|| \neq 0, i=1,\ldots,n-1\}$.*

Proof: Let $D_\lambda^* \subseteq \{L(X''_{n-\lambda}) \oplus 1, L(X''_{n-\lambda}) \in D_\lambda\}$, and $D_i^* = \{0\}, (i \neq \lambda)$. Moreover, let

$$g(X) = g(X'_i, X''_{n-i}) = \sum_{i=1}^{n-1} \sum_{\sigma^{(i)}=(\sigma^{(i)}_{j_1},\ldots\sigma^{(i)}_{j_i})\in B_i} \left(\prod_{s=j_1}^{j_i} (x_s \oplus \sigma^{(i)}_s \oplus 1) \right) \cdot g(\sigma^{(i)}, X''_{n-i}) \quad (5.2)$$

where $g(\sigma^{(i)}, X''_{n-i}) \in D^*_i$, and $||B_i|| \neq 0$. Note that $f(\sigma^{(i)}, X''_{n-i}) \cdot g(\sigma^{(i)}, X''_{n-i}) = 0$ for all $(X'_i, X''_{n-i}) \in GF(2)^n$. It is easily verified that $f \cdot g = 0$ and $\deg(g) \leq \lambda + 1$, where $\lambda = \min\{i \mid ||B_i|| \neq 0, i = 1, \ldots, n-1\}$.

■

Example 5.1.7 *Consider an $n = 8$ variable Boolean function $f(x)$ whose truth table is given below. Using the existing algorithm in [3], we can easily calculate the exact AI value of this function, getting $AI = 2$. On the other hand, using our algorithm we find a decomposition for this function, where $||B_2|| = 1, ||B_4|| = 4, ||B_6|| = 32, ||B_i|| = 0, i \neq (2, 4, 6)$. Using Theorem 5.1.6, to estimate the theoretical upper bound on AI value, we found $AI \leq 3, (\lambda + 1 = 2 + 1 = 3)$, which is consistent to the exact value $AI = 2$.*

0001000100010001000100010100010000010001000100010001000101000100000
1000100010001000100010100010000100010001000100010001001110111000100
0100010100000100010100000100010001000101000001000101000001000100010
0010100000100010100000100100010001001110010001001110010

Note that the number of elements in the set of affine subfunctions on $(n-i)$-variable is $||B_i|| \times 2^{n-i}$ (for those a_i for which g_i passes the linearity test) over $GF(2)^n$, for $i = 1, \ldots, n-1$. It is clear that $||B_i|| \times 2^{n-i}$ will be relatively large if i is relatively small and $||B_i|| \neq 0$. To estimate the maximal size of $||B_i|| \neq 0$ for small i, we propose an optimized algorithm below. In difference to Algorithm 1, where a fixed decomposition of $n-1$ variables gives unique (fixed) sets B_i, Algorithm 2 selects the best decomposition in terms of maximal cardinality of B_i. It implies that in each step we select a decomposition which for a fixed choice of the positions of input variables gives maximal number of affine subfunctions.

Algorithm 2 (Optimized Partial Linear Decomposition)

Step 1 For a given n-variable Boolean function $f \in \mathcal{B}_n$, let $k = \lceil \log_2 n \rceil$. Set counters $T^j_{B_i} = 0$, and tables C^j_t for storing $a_i \in B_i$, where $i = 1, \ldots, n-1$ and $j = 1, \ldots, n$.

Step 2 For each $x_j = a_1 \in GF(2)$, $j = 1, \ldots, n$, randomly choose different 2^k pairs $(\alpha^\ell_{n-1}, \beta^\ell_{n-1})$, where $\alpha^\ell_{n-1}, \beta^\ell_{n-1} \in GF(2)^{n-1}$. Let $g = f_{x_j=a_1}(x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n)$ and for each pair $(\alpha^\ell_{n-1}, \beta^\ell_{n-1})$, $\ell = 1, \ldots, 2^k$, perform the linear relation test below:

$$g_1(\alpha^\ell_{n-1} \oplus \beta^\ell_{n-1}) = g_1(\alpha^\ell_{n-1}) \oplus g_1(\beta^\ell_{n-1}) \oplus g_1(0, \ldots, 0).$$

(1) If all 2^k pairs $(\alpha^\ell_{n-1}, \beta^\ell_{n-1})$ can pass through the linear test, then let $T^j_{B_1} = T^j_{B_1} + 1$. Otherwise, save the corresponding a_1 to table C^j_1.

(2) Let $I_1 = \{j \mid \max_{j=1}^{n} T_{B_1}^j\}$.

Step 3 Randomly choose $j_1^* \in I_1$, for each $j \in \{1, \ldots, n\}, (j_1^* \neq j)$ and for each $(a_1, x_j) = a_2 \in GF(2)^2, a_1 \in C_1^{j_1^*}$, randomly choose 2^k pairs $(\alpha_{n-2}^\ell, \beta_{n-2}^\ell)$, where $\alpha_{n-2}^\ell, \beta_{n-2}^\ell \in GF(2)^{n-2}$. Let $g_2 = f_{(x_{j_1^*}, x_j)=a_2}$ and for each pair $(\alpha_{n-2}^\ell, \beta_{n-2}^\ell)$, $\ell = 1, \ldots, 2^k$, perform the linear relation test below:

$$g_2(\alpha_{n-2}^\ell \oplus \beta_{n-2}^\ell) = g_2(\alpha_{n-2}^\ell) \oplus g_2(\beta_{n-2}^\ell) \oplus g_1(0, \ldots, 0).$$

(1) If all 2^k pairs $(\alpha_{n-2}^\ell, \beta_{n-2}^\ell)$ pass the linear test, then let $T_{B_2}^j = T_{B_2}^j + 1$. Otherwise, save the corresponding a_2 to table C_2^j.

(2) Let $I_2 = \{j \mid \max_{j=1, j\neq j^*}^{n} T_{B_2}^j\}$.

Step 4 Similarly, repeat the Step 3 above, by increasing the size of input variables, i.e., $(a_{i-1}, x_j) = a_i \in GF(2)^i$, and $(i = 3, \ldots, n-k), j \in \{1, \ldots, n\}, j \neq j_t^*, j_t^* \in I_t, t = (1, \ldots, i-1)$. In general, for $i = (n-k+1, \ldots, n-2)$, randomly choose different 2^{n-i} pairs $(\alpha_{n-i}^\ell, \beta_{n-i}^\ell)$, where $\alpha_{n-i}^\ell, \beta_{n-i}^\ell \in GF(2)^{n-i}$, $\ell = 1, \ldots, 2^{n-i}$, and check whether all 2^{n-i} pairs can pass the linear relation test or not, where $g_i = f_{(x_{j_{i-1}^*}, x_j)=a_i}$.

Step 5 Return the values of $||B_i|| = T_{B_i^{j_t^*}}$, for $i, t = 1, \ldots, n-1$.

Similarly to the analysis of Algorithm 1, the success rate of this algorithm is also about $p = 1$. Step 2 requires about $2 \times 2^k \times \binom{n}{1}$ operations, whereas Step 3 needs about $2^2 \times 2^k \times \binom{n-1}{1}$ operations. The time complexity of this algorithm is dominated by Step 2-4, i.e.,

$$T_{complexity} = \sum_{i=1}^{n-k} 2^i \times 2^k \times \binom{n+1-i}{1} + \sum_{j=n-k+1}^{n-2} 2^j \times 2^{n-j} \times \binom{n+1-j}{1}.$$

Moreover, we have

$$\begin{aligned} T_{complexity} &= 2^k \sum_{i=1}^{n-k} 2^i \times (n+1-i) + \sum_{j=n-k+1}^{n-1} 2^j \times 2^{n-j} \times (n+1-j) \\ &< (2^n - 2^k + 2^n \times (n-1-(n-k+1)+1)) \times n \\ &< (k \times 2^n) \times n, \end{aligned}$$

where $k = \log_2 n$. Therefore, the time complexity of this algorithm is about $O(n2^n \times \log_2 n)$ operations. The memory complexity is only about $O(n2^{n-1})$ bits, which is mainly used to save the tables C_t^j, for $t = 1, \ldots, n-1, j = 1, \ldots, n$. Notice also that Theorem 5.1.2 is valid for Algorithm 2, thus an upper bound on AI can be derived using either Algorithm 1 or Algorithm 2.

Remark 5.1.8 *One may notice that both Algorithms 1 and 2 start with fixing one coordinate. In the case of highly nonlinear functions we do not expect to get affine subfunctions by fixing some small number of variables. Therefore, one may run these algorithms backwards, i.e., to start with a selection of $n-2$ or $n-3$ fixed coordinates. By fixing, say $n-2$ coordinates, it is quite likely that we get many affine subfunctions.*

However, further selection of fixed coordinates for sets B_i ($i < n-2$) is highly affected by certain complicated properties of these sets which are not mentioned in Corollary 5.1.3. Thus, finding an explicit non-probabilistic algorithm which provides a complete description of sets B_i, which result in a decomposition (5.1) of an arbitrary input function f, is left as an open problem. Note that the existence of decomposition (5.1) of an arbitrary function f is guaranteed by Theorem 5.1.2.

5.2 Estimating the resistance against AA and FAA

In this section, the resistance of Boolean functions against (fast) algebraic attack is discussed. The fact that our algorithms provide an upper bound on AI is not sufficient for efficient estimation of the entire algebraic properties of a given function. Indeed, in the first place an estimate of a lower bound on AI is of even greater importance but also a tight lower and upper bound concerning the algebraic degree of $\deg(g)+\deg(h)$ in the relation $fg = h$ are necessary. These bounds are derived in this section (through the set of conditions relating the main decomposition parameters) which then along with the use of Algorithm 2 gives us an efficient algorithm for estimating the algebraic properties of a given function.

5.2.1 Resistance to AA

Without loss of generality, we assume $X = (x_1, \ldots, x_n) \in GF(2)^n$, $X'_i = (x_1, \ldots, x_i) \in GF(2)^i$, $X''_{n-i} = (x_{i+1}, \ldots, x_n) \in GF(2)^{n-i}$, and then the equality (5.1) has the form below:

$$f(X'_i, X''_{n-i}) = \sum_{i=1}^{n-1} \sum_{\sigma^{(i)}=(\sigma_1^{(i)},\ldots,\sigma_i^{(i)})\in B_i} \prod_{l=1}^{i} (x_l \oplus \sigma_l^{(i)} \oplus 1) \cdot f(\sigma^{(i)}, X''_{n-i}), \qquad (5.3)$$

where $f(\sigma^{(i)}, X''_{n-i}) \in D_i$, $X'_i \in B_i$, and $||B_i|| \neq 0$. Here, again $D_i \subseteq \{L(X''_{n-i}) \mid L(X''_{n-i}) = c \cdot X''_{n-i} \oplus b, c \in GF(2)^{n-i}, b \in GF(2)\}$.

Note that any annihilator of f can be represented as

$$g^*(X'_i, X''_{n-i}) = \sum_{i=1}^{n-1} \sum_{\sigma^{(i)}=(\sigma_1^{(i)},\ldots,\sigma_i^{(i)})\in B_i} \prod_{l=1}^{i} (x_l \oplus \sigma_l^{(i)} \oplus 1) \cdot u_{[\sigma^{(i)}]}(X''_{n-i}), \qquad (5.4)$$

where $u_{[\sigma^{(i)}]}(X''_{n-i})$ is any annihilator of $f(\sigma^{(i)}, X''_{n-i})$, i.e., $u_{[\sigma^{(i)}]}(X''_{n-i}) \cdot f(\sigma^{(i)}, X''_{n-i}) = 0$, $\sigma^{(i)} \in B_i$.

Let us restrict the degree of g^* to a fixed value $r + d \leq n/2$. If we need to cancel the terms in the ANF of g^* containing $x_{j_1} \cdots x_{j_q}$ for any q in the range $d - 1 < q \leq i < n$, where d is a fixed integer and $\{j_1, \cdots, j_q\} \subset \{1, \cdots, i\}$, then the sufficient condition is that,

$$\sum_{i=1}^{n-1} \sum_{\sigma\in B_i} u_{[\sigma^{(i)}]}(X''_{n-i}) = \sum_{i=1}^{n-1} \sum_{\sigma\in B_i} (f(\sigma^{(i)}, X''_{n-i}) \oplus 1) \times u'_{i,[\sigma]}(X''_{n-i}) = 0, \qquad (5.5)$$

where each $u'_{[\sigma^{(i)}]}(X''_{n-i})$, for any $\sigma^{(i)} \in B_i$, is at most of degree r_i given by,

$$\begin{aligned} u'_{[\sigma^{(i)}]}(X''_{n-i}) &= a_0^{\sigma^{(i)}} \oplus a_1^{\sigma^{(i)}} x_{i+1} \oplus \cdots \oplus a_{n-i}^{\sigma^{(i)}} x_n \oplus \cdots \\ &\oplus a_{1,\ldots,r_i}^{\sigma^{(i)}} x_{i+1} \cdots x_{i+r_i} \oplus \cdots \oplus a_{n-r_i+1,\ldots,n}^{\sigma^{(i)}} x_{n-r_i+1} \cdots x_n. \quad (5.6) \end{aligned}$$

Note that $\deg(f(\sigma^{(i)}, X''_{n-i})) \leq 1$, for any $\sigma^{(i)} \in B_i$.

Then we try to select the coefficients $a_\iota^\sigma \in GF(2)$ in (5.6) in such a way that the terms in the ANF of g^* containing $x_{j_1} \cdots x_{j_q}$ are all cancelled for any q in the range $d-1 < q \leq i < n$, where d is a fixed integer and $\{j_1, \cdots, j_q\} \subset \{1, \cdots, i\}$. This also implies that the degree of

$$\sum_{\sigma^{(i)}=(\sigma_1^{(i)},\ldots\sigma_i^{(i)})\in B_i} \prod_{l=1}^{i} (x_l \oplus \sigma_l^{(i)} \oplus 1) \times (f(\sigma^{(i)}, X''_{n-i}) \oplus 1) \times u'_{[\sigma]}(X''_{n-i}),$$

is at most $r_i + d, (1 \leq i < n-1)$ since $\deg(u'_{[\sigma^{(i)}]}(X''_{n-i}))+ \deg(f(\sigma^{(i)}, X''_{n-i}))+ \deg(x_{j_1} \cdots x_{j_q}) \leq r_i + 1 + d - 1 = r_i + d$, for $\sigma \in B_i$. If we are able to obtain such a choice of the coefficients $a_\iota^{\sigma^{(i)}} \in GF(2)$ in $u'_{[\sigma^{(i)}]}(X''_{n-i})$, then the degree of g^* would be at most $\max_{i=0}^{n-1}\{r_i\} + d$ (but also at least $\min_{i=0}^{n-1}\{r_i\} + d$), for all $B_i, (1 \leq i < n-1)$. Notice that when $\sigma^{(i)}$ runs through all $B_i, (1 \leq i < n-1)$, we obtain in total $\sum_{\{1\leq i\leq n-1, ||B_i||\neq 0\}} ||B_i|| \times \sum_{j=0}^{r_i} \binom{n-i}{j}$ unknown coefficients $a_\iota^{\sigma^{(i)}} \in GF(2)$. On the other hand, cancelling all the terms in the ANF of g^* containing $x_{j_1} \cdots x_{j_q}$, will induce certain restrictions on the coefficients $a_\iota^{\sigma^{(i)}} \in GF(2)$ in the resulting system of homogeneous linear equations (involving these $a_\iota^{\sigma^{(i)}} \in GF(2)$) whose total number is given by,

$$\sum_{\{1\leq i\leq n-1, ||B_i||\neq 0\}} \sum_{l=0}^{i-d} \binom{i}{i-l} \sum_{j=0}^{r_i+1} \binom{n-i}{j}, \quad (5.7)$$

where $i - d \geq 0, i - l \geq 0$. The binomial sum term $\sum_{l=0}^{i-d} \binom{i}{i-l}$ in the above equation refers to counting all the terms that contain $x_{j_1} \cdots x_{j_q}$ in $\sum_{\sigma^{(i)}\in B_i} \Pi_{l=0}^{i}(x_i \oplus \sigma_i^{(i)} \oplus 1)$ section (where $\{j_1, \cdots, j_q\} \subset \{1, \cdots, i\}$), whose degree q is in the range d to i. The binomial sum term $\sum_{j=0}^{r_i+1} \binom{n-i}{j}$ counts all the possible terms that are involved in the $f(\sigma^{(i)}, X''_{n-i}) \times u'_{[\sigma^{(i)}]}(X''_{n-i})$ portion. Moreover, the summation (i.e., $\sum_{\{1\leq i\leq n-1, ||B_i||\neq 0\}}$) in the above equation takes into account all homogeneous linear equations for all $||B_i|| \neq 0, (1 \leq i < n-1)$.

It is obvious that there will be solutions to this homogeneous system of equations if the number of equations is less or equal than the number of unknowns.

Thus, if the condition below is satisfied, then we will obtain at least one Boolean function g^* of degree $r' + d \leq \deg(g^*) \leq r + d$ (by solving the system for unknown $a_\iota^{\sigma^{(i)}} \in GF(2)$) with $r' + d \leq \deg(g^*) \leq r + d$, where $r = \max_{i=0}^{n-1}\{r_i\}$ and $r' = \min_{i=0}^{n-1}\{r_i\}$. This gives us both a lower and upper bound on the value of AI and these bounds appear to be tight for randomly selected Boolean functions.

Condition 0:

$$\sum_{\{1\le i\le n-1,||B_i||\neq 0\}} ||B_i|| \times \sum_{j=0}^{r_i} \binom{n-i}{j} \geq$$
$$\sum_{\{1\le i\le n-1,||B_i||\neq 0\}} (\sum_{l=0}^{i-d} \binom{i}{i-l} \times \sum_{j=0}^{r_i+1} \binom{n-i}{j}) \tag{5.8}$$

It seems to be difficult to obtain a concise expression for the optimal choice of the parameters r', r and d.

Remark 5.2.1 *From the inequality (5.8), the cardinalities $||B_i||$, $i = 1, \ldots, n-1$, affect the AI value. In particular, larger $||B_i||$ and smaller i usually implies smaller AI.*

5.2.2 Resistance to FAA

Our main objective now is to confirm the existence of a low degree Boolean function g' such that the function fg' also has a low degree, though more precisely the actual goal is to minimize $\deg(g') + \deg(fg')$, where f has the form given by (5.3). Let us restrict the degree of g' with a fixed value $r + s$, by considering

$$g'(X'_i, X''_{n-i}) = \sum_{i=1}^{n-1} \{ \sum_{\sigma^{(i)}=(\sigma_1^{(i)},\ldots\sigma_i^{(i)})\in B_i} \prod_{l=1}^{i} (x_l \oplus \sigma_l^{(i)} \oplus 1) \cdot \xi_{[\sigma^{(i)}]}(X''_{n-i}) \}, \tag{5.9}$$

where each $\xi_{[\sigma^{(i)}]}(X''_{n-i})$, for any $\sigma^{(i)} \in B_i$, is a degree r_i function given by,

$$\begin{aligned} \xi_{[\sigma^{(i)}]}(X''_{n-i}) &= b_0^{\sigma^{(i)}} \oplus b_1^{\sigma^{(i)}} x_{i+1} \oplus \cdots \oplus b_{n-i}^{\sigma^{(i)}} x_n \oplus \cdots \\ &\oplus b_{1,\ldots,r_i}^{\sigma^{(i)}} x_{i+1} \cdots x_{i+r_i} \oplus \cdots \oplus b_{n-r_i+1,\ldots,n}^{\sigma^{(i)}} x_{n-r_i+1} \cdots x_n. \end{aligned} \tag{5.10}$$

There are two basic conditions that need to be satisfied so that both g' and fg' are of low degree.

(1) Firstly, we need to specify g' to be of low algebraic degree. We try to select the coefficients $b_\iota^{\sigma^{(i)}} \in GF(2)$ in (5.10) in such a way that the terms in the ANF of g' containing $x_{j_1} \cdots x_{j_q}$ are all cancelled for any q in the range $s < q \leq i < n$, where s is a fixed integer and $\{j_1, \cdots, j_q\} \subset \{1, \cdots, i\}$. This also implies that the degree of

$$\sum_{\sigma^{(i)}=(\sigma_1^{(i)},\ldots\sigma_i^{(i)})\in B_i} \prod_{l=1}^{i} (x_l \oplus \sigma_l^{(i)} \oplus 1) \cdot \xi_{[\sigma^{(i)}]}(X''_{n-i})$$

would be at most $r_i + s$ for each B_i and $1 \leq i < n-1$. If we are able to obtain such a choice of the coefficients $b_\iota^{\sigma^{(i)}} \in GF(2)$ in $\xi_{[\sigma^{(i)}]}(X''_{n-i})$, then the degree of g' would be at most $\max_{i=0}^{n-1}\{r_i\} + s$ (but also at least $\min_{i=0}^{n-1}\{r_i\} + s$), for all

B_i. Notice that when $\sigma^{(i)}$ runs through all B_i, $1 \leq i < n-1$, we obtain in total $\sum_{\{1\leq i\leq n-1,||B_i||\neq 0\}} ||B_i|| \times \sum_{j=0}^{r_i} \binom{n-i}{j}$ unknown coefficients $b_\iota^{\sigma^{(i)}} \in GF(2)$. On the other hand, to cancel all the terms in the ANF of g' containing $x_{j_1} \cdots x_{j_q}$, will induce certain restrictions on the coefficients $b_\iota^{\sigma^{(i)}} \in GF(2)$ in the resulting system of homogeneous linear equations (involving these $b_\iota^{\sigma^{(i)}} \in GF(2)$) whose total number is given by,

$$\sum_{\{1\leq i\leq n-1,||B_i||\neq 0\}} (\sum_{l=0}^{i-s-1} \binom{i}{i-l} \sum_{j=0}^{r_i} \binom{n-i}{j}), \tag{5.11}$$

where $i-s-1 \geq 0, i-l \geq 0$. The binomial sum term $\sum_{l=0}^{i-s-1} \binom{i}{i-l}$ in the above equation refers to counting all the terms containing $x_{j_1} \cdots x_{j_q}$ in $\sum_{\sigma\in B_i} \Pi_{l=0}^{i}(x_i \oplus \sigma_i \oplus 1)$ section (where $\{j_1, \cdots, j_q\} \subset \{1, \cdots, i\}$), whose degree q is in the range $s+1$ to i. The binomial sum term $\sum_{j=0}^{r_i} \binom{n-i}{j}$ counts all the possible terms that are involved in the $\xi_{[\sigma^{(i)}]}(X''_{n-i})$ portion. Moreover, the summation (i.e., $\sum_{\{1\leq i\leq n-1,||B_i||\neq 0\}}$) in the above equation takes into account all homogeneous linear equations for all $B_i \neq \emptyset, (1 \leq i < n-1)$.

Thus, if the

Condition 1:

$$\sum_{\{1\leq i\leq n-1,||B_i||\neq 0\}} ||B_i|| \times \sum_{j=0}^{r_i} \binom{n-i}{j} \geq \sum_{\{1\leq i\leq n-1,||B_i||\neq 0\}} \Big(\sum_{l=0}^{i-s-1} \binom{i}{i-l} \sum_{j=0}^{r_i} \binom{n-i}{j} \Big),$$

i.e.,

$$\sum_{\{1\leq i\leq n-1,||B_i||\neq 0\}} ||B_i|| \geq \sum_{\{1\leq i\leq n-1,||B_i||\neq 0\}} \sum_{l=0}^{i-s-1} \binom{i}{i-l} \tag{5.12}$$

is satisfied, then we will obtain at least one Boolean function g' (by solving the system for unknown $b_\iota^\sigma \in GF(2)$) with $r'+s \leq \deg(g') \leq r+s$, where $r = \max_{i=0}^{n-1}\{r_i\}$ and $r' = \min_{i=0}^{n-1}\{r_i\}$.

(2) Secondly, we note that

$$f(X'_i, X''_{n-i}) \times g'(X'_i, X''_{n-i}) = \sum_{i=1}^{n-1} \sum_{(\sigma_1^{(i)},\ldots\sigma_i^{(i)})\in B_i} \prod_{l=1}^{i} (x_l \oplus \sigma_l^{(i)} \oplus 1) \times f(\sigma^{(i)}, X''_{n-i}) \times \xi_{[\sigma^{(i)}]}(X''_{n-i}),$$

where each $\deg(f(\sigma^{(i)}, X''_{n-i})) \leq 1$, for any $\sigma^{(i)} \in B_i$, $(1 \leq i \leq n-1)$.
It is clear that the algebraic degree of

$$\sum_{\sigma^{(i)}=(\sigma_1^{(i)},\ldots\sigma_i^{(i)})\in B_i} \prod_{l=1}^{i} (x_l \oplus \sigma_l^{(i)} \oplus 1) \times f(\sigma^{(i)}, X''_{n-i}) \times \xi_{[\sigma^{(i)}]}(X''_{n-i}) \tag{5.13}$$

is at most $r_i + i + 1$, due to the fact that the degree of $\xi_{[\sigma^{(i)}]}(X''_{n-i})$ is r_i and the degree of

$$\sum_{\sigma^{(i)}=(\sigma_1^{(i)},\ldots\sigma_i^{(i)})\in B_i} \prod_{l=1}^{i} (x_l \oplus \sigma_l^{(i)} \oplus 1) \times f(\sigma^{(i)}, X''_{n-i})$$

is at most $i+1$. Moreover, to restrict the degree of fg' not to be larger than e, we require that the ANF of fg' contains the terms of algebraic degree at most e. In other words, in the ANF of fg', the coefficients of the terms of algebraic degree greater than e must be equal to zero. Notice that the coefficients of these terms can be expressed as some linear equations of the unknowns $b_\iota^{\sigma^{(i)}} \in GF(2), \sigma^{(i)} \in B_i$, $1 \leq i \leq n-1$, which in total induces at most Λ_1 equations, where

$$\Lambda_1 = \sum_{\{1\leq i\leq n-1, ||B_i||\neq 0\}} \Big(\sum_{j=e+1}^{i+r_i+1} \binom{n}{j} - \sum_{l_1=e+1}^{i+r_i+1} \binom{i}{l_1-v_1} \sum_{v_1=r_i+2}^{n-i} \binom{n-i}{v_1} \Big),$$

$l_1 - v_1 \geq 0$, and $l_2 - v_2 \geq 0$.

The sum(i.e., $\sum_{\{1\leq i\leq n-1, ||B_i||\neq 0\}}$) in Λ_1 refers to counting all homogeneous linear equations for all $B_i \neq \emptyset, (1 \leq i < n-1)$. The binomial sum term $\sum_{j=e+1}^{i+r_i+1} \binom{n}{j}$ in Λ_1 stands for all the terms in the ANF of equality (5.13) whose degree (denoted by j) ranges from $e+1$ to $i+r_i+1$. From this part we have to subtract those equations, corresponding to the double binomial sum (i.e., $\sum_{l_1=e+1}^{i+r_i+1} \binom{i}{l_1-v_1} \sum_{v_1=r_i+2}^{n-i} \binom{n-i}{v_1}$), that cannot appear in the ANF in equality (5.13). The first binomial term of the double binomial sum takes into account the number of terms of the form $x_{j_1} \cdots x_{j_{v_1}}$, where $r_i + 2 \leq v_1 \leq n-i$ and $i+1 \leq j_1 < j_2 < \cdots < j_{v_1} \leq n$, since clearly $f(\sigma^{(i)}, X''_{n-i}) \times \xi_{[\sigma^{(i)}]}(X''_{n-i})$ in equality (5.13) is of degree at most $r_i + 1$ in $x_{i+1}, \ldots, x_n$. The second binomial term of the double binomial sum takes care of the number of terms of the form $x_{j_1^*} \cdots x_{j_{l_1-v_1}^*}$ as a constituent part of non-appearing terms of the form $x_{j_1} \cdots x_{j_{v_1}} x_{j_1^*} \cdots x_{j_{l_1-v_1}^*}$, where $l_1 - v_1 > 0, e+1 \leq l_1 \leq r+1+i$ and $1 \leq j_1^* < j_2^* < \cdots < j_{l_1-v_1}^* \leq i$, since clearly $\sum_{\sigma^{(i)}=(\sigma_1^{(i)},\ldots\sigma_i^{(i)})\in B_i} \prod_{l=1}^{i}(x_l \oplus \sigma_l^{(i)} \oplus 1)$ in equality (5.13) is of degree at most i in $x_1, \ldots, x_i$.

Therefore, we will obtain at least one Boolean function g' such that the degree of fg' is e if Condition 2 below is satisfied.

Condition 2:

$$\sum_{\{1\leq i\leq n-1, ||B_i||\neq 0\}} ||B_i|| \times \sum_{j=0}^{r_i} \binom{n-i}{j} \geq \Lambda_1, \tag{5.14}$$

where

$$\Lambda_1 = \sum_{\{1\leq i\leq n-1, ||B_i||\neq 0\}} \Big(\sum_{j=e+1}^{i+r_i+1} \binom{n}{j} - \sum_{l_1=e+1}^{i+r_i+1} \binom{i}{l_1-v_1} \sum_{v_1=r_i+2}^{n-i} \binom{n-i}{v_1} \Big), \tag{5.15}$$

$l_1 - v_1 \geq 0$.

5.2.3 An algorithm for estimating the resistance against AA and FAA

In this section, an algorithm for estimating the resistance of Boolean functions against both AA and FAA is introduced. It uses previously described algorithms for

finding a good decomposition of a Boolean function, thus the sets B_i are found by using either Algorithm 2 or Algorithm 1.

Algorithm 3
Step 1 For a given n-variable Boolean function f, use Algorithm 2 (or Algorithm 1) to calculate the values of $||B_i||$, for $i = 1, \ldots, n-1$.
Step 2 Use Conditions $0-2$ to calculate

$$\triangle_{AA}^{lower} = \min\{\lambda + 1, \lceil n/2 \rceil, r' + d\}, \;\; \nabla_{AA}^{upper} = \min\{\lceil n/2 \rceil, r + d\},$$

and

$$\triangle_{FAA}^{lower} = \min\{n-1, r' + s + e\}, \;\; \nabla_{FAA}^{upper} \min\{n-1, r + s + e\}$$

for AA and FAA, where $\lambda = \min\{i \mid ||B_i|| \neq 0, i = 1, \ldots, n-1\}$.
Step 3 Repeat Step 1 and Step 2 $[\frac{n}{\log_2 n}]$ times, and return the minimum values of upper and lower bound: $(\triangle_{AA}^{lower}, \nabla_{AA}^{upper})$ and $(\triangle_{FAA}^{lower}, \nabla_{FAA}^{upper})$, respectively.

The time complexity of this algorithm is about $O([\frac{n}{\log_2 n}] \times \log_2 n \times n2^n) \approx O(n^2 2^n)$ operations, if Algorithm 2 is used to search for the values of $||B_i||$, $i = 1, \ldots, n-1$. The memory complexity is about $O(n2^n)$ bits.

Remark 5.2.2 *Algorithm 3 only gives a theoretical upper and lower bound on both AI and $r+s+e$ for FAA. On the other hand, although Algorithm 2 proposes an approach for calculating the maximum $||B_i|| \neq 0$ for small i, an optimal decomposition for B_i in Algorithm 3 is still an open problem.*

Table 5.1: The time complexity of our algorithm versus previous works.

The ability against AA or FAA	The time complexity	Resource
AA	$O(D^3)$	[25]
AA	$O(n^d)$	[30]
AA	$O(D^2)$	[3]
FAA	$O(D^2E)$	[3]
FAA	$O(DE^2 + D^2)$	[8]
AA or FAA	$O(n^{2n}D)$	[29]
AA or FAA	$O(D^{2\pm\varepsilon})$	[56]
AA or FAA	$O(n^2 2^n)$	new

$(D = \sum_{i=0}^{d} \binom{n}{i}, E = \sum_{i=0}^{e} \binom{n}{i}, \varepsilon = 0.5)$.

Table 1 describes the time complexity of previous works and of our algorithm for estimating the resistance of random n-variable Boolean functions against AA and FAA. In particular, Table 2 describes a comparison of the time complexity for $30 \leq n \leq 40$. For instance, for $n = 40$, the best previous known time complexity is about $2^{55.51}$ operations in [56]. However, the time complexity of our algorithm is only about $2^{28.64}$ operations. It is evident that our new algorithm has a more favorable time complexity than other methods though being probabilistic it may not succeed in outputting the best possible decomposition choice which may result in a somewhat lose lower and upper bound.

Table 5.2: A time complexity comparison for $30 \leq n \leq 40$.

n	[25]	[30]	[8]§	[29]	[56]†	[56] ‡	new
30	$2^{81.63}$	$2^{73.60}$	$2^{54.42}$	$2^{144.42}$	$2^{68.02}$	$2^{40.81}$	$2^{22.81}$
31	$2^{84.49}$	$2^{74.31}$	$2^{56.33}$	$2^{145.37}$	$2^{70.41}$	$2^{42.24}$	$2^{22.81}$
32	$2^{87.49}$	$2^{80.00}$	$2^{58.33}$	$2^{157.16}$	$2^{72.91}$	$2^{43.74}$	$2^{24.00}$
33	$2^{90.36}$	$2^{80.71}$	$2^{60.24}$	$2^{158.12}$	$2^{75.30}$	$2^{45.18}$	$2^{24.00}$
34	$2^{93.36}$	$2^{86.49}$	$2^{62.24}$	$2^{171.09}$	$2^{77.80}$	$2^{46.68}$	$2^{25.17}$
35	$2^{96.24}$	$2^{87.20}$	$2^{64.16}$	$2^{172.05}$	$2^{80.20}$	$2^{48.12}$	$2^{25.17}$
36	$2^{99.24}$	$2^{93.06}$	$2^{66.16}$	$2^{183.20}$	$2^{82.70}$	$2^{49.62}$	$2^{26.34}$
37	$2^{102.12}$	$2^{93.77}$	$2^{68.08}$	$2^{184.16}$	$2^{85.10}$	$2^{51.06}$	$2^{26.34}$
38	$2^{105.12}$	$2^{99.71}$	$2^{70.08}$	$2^{196.46}$	$2^{87.60}$	$2^{52.56}$	$2^{27.50}$
39	$2^{108.01}$	$2^{100.42}$	$2^{72.01}$	$2^{197.42}$	$2^{90.01}$	$2^{54.01}$	$2^{27.50}$
40	$2^{111.01}$	$2^{106.44}$	$2^{74.01}$	$2^{209.88}$	$2^{92.51}$	$2^{55.51}$	$2^{28.64}$

(§ : $e = 1$, † : $D^{2+\varepsilon}$, ‡ : $D^{2-\varepsilon}$)

Example 5.2.3 *Choose an $n = 12$ variable Boolean function $f(x)$ whose truth table in the hexadecimal format is given below. Using the algorithms in [3], we could easily verify the actual resistance of this function against AA and FAA to be $AI(f) = 5$, $deg(g)+deg(h) \geq 7$, ($deg(f) = 8$) for nonzero Boolean functions g and h satisfying $fg = h$. On the other hand, we could obtain a decomposition of this function given by $||B_6|| = 13, ||B_7|| = 102, ||B_i|| = 0, i \neq (6, 7)$, when using the canonical order of fixing the input variables (thus $(x_1 \rightarrow x_2 \ldots \rightarrow x_{11})$). Algorithm 3 then gives the following estimates of the lower and upper bound on AA: $5 \leq AI(f) \leq 6$, $(\triangle_{AA}^{lower} = r' + d = 1 + 4 = 5, \nabla_{AA}^{upper} = r + d = 1 + 5 = 6)$, which is consistent to the actual value $AI(f) = 5$. Moreover, another decomposition of this function is given by $||B_9|| = 512, ||B_i|| = 0, i \neq 9$, if the order of fixing the input variables is $(x_1, x_2, x_3, x_{10}, x_{11}, x_{12}, x_4, x_5, x_6, x_7, x_8, x_9)$. Using Algorithm 3 to estimate the lower and upper bound regarding the resistance of f against FAA gives $6 \leq deg(g)+deg(h) \leq 7$, $(\triangle_{FAA}^{lower} = r'+s+e = 0+1+5 = 6, \nabla_{FAA}^{upper} r+s+e = 1+1+5 = 7)$, which is also consistent to the actual lower bound $deg(g)+deg(h) \geq 7$.*

6666 9999 6666 6666 6666 6666 6666 9999 6666 9999 6666 6666 9999 9999 9999 6666 6699 6699 6699 9966 6699 6699 6699 9966 9966 9966 6699 9966 6699 6699 9966 6699 33cc 33cc 33cc cc33 33cc 33cc 33cc cc33 cc33 33cc cc33 cc33 33cc cc33 33cc 33cc 0f0f f0f0 0f0f f0f0 00ff ff00 00ff ff00 0f0f 0f0f f0f0 f0f0 00ff 00ff ff00 ff00 55aa aa55 55aa 55aa 55aa aa55 55aa 55aa 55aa 55aa 55aa aa55 aa55 aa55 aa55 55aa 6996 6996 9669 6996 6996 6996 9669 6996 6996 6996 9669 6996 9669 9669 6996 9669 6969 6969 6969 9696 9696 9696 6969 9696 6969 6969 6969 9696 6969 6969 9696 6969 0000 ffff ffff 0000 55aa 55aa 5555 aaaa 6666 6666 5a5a 5a5a 3c3c 3c3c 33cc 33cc 5a5a 5a5a 5a5a a5a5 5a5a 5a5a 5a5a a5a5 5a5a 5a5a 5a5a a5a5 a5a5 a5a5 a5a5 5a5a 3cc3 c33c 3cc3 3cc3 3cc3 c33c 3cc3 3cc3 3cc3 c33c 3cc3 3cc3 c33c 3cc3 c33c c33c 0ff0 0ff0 f00f 0ff0 0ff0 0ff0 0ff0 f00f 0ff0 0ff0 f00f 0ff0 f00f f00f f00f 0ff0 5555 aaaa aaaa 5555 0f0f f0f0 f0f0 0f0f 3333 cccc cccc 3333 00ff ff00 ff00 00ff 3c3c 3c3c 3c3c c3c3 c3c3 3c3c c3c3 c3c3 3c3c 3c3c 3c3c c3c3 3c3c c3c3 3c3c 3c3c 5555 aaaa 5555 aaaa 3333 cccc 3333 cccc 5555 5555 aaaa aaaa 3333 3333 cccc cccc 5aa5 5aa5 a55a 5aa5 5aa5 5aa5 a55a 5aa5 5aa5 5aa5 5aa5 a55a a55a a55a a55a 5aa5 3333 cccc 0ff0 0ff0 3cc3 c33c 6699 9966 0f0f f0f0 6969 6969 5aa5 a55a 6996 9669

Example 5.2.4 *We select a random Boolean function $f(x) \in \mathcal{B}_{14}$ whose truth table is given in the Appendix. Similarly, using algorithms in [3], we easily verified that the actual resistance of this function against FAA is $deg(g)+deg(h) \geq 13$, ($deg(f) = 14$) for nonzero Boolean functions g and h such as $fg = h$. On the other hand, we obtained a decomposition for this function, i.e., $||B_{11}|| = 132, ||B_{12}|| = 1761, ||B_{13}|| = 4142, ||B_i|| = 0, i \neq (11, 12, 13)$. Using Algorithm 3 to estimate the theoretical lower bound on the ability against FAA, we found $r'+s+e = 0+6+7 = 13$, and $\triangle_{FAA}^{lower} = \nabla_{FAA}^{upper} = 13$, which is also completely consistent to the actual value $deg(g)+deg(h) \geq 13$.*

Remark 5.2.5 *Some simulations for randomly chosen Boolean functions $f(x)$ with $n = 14$ variables, were also performed using Algorithm 3. We found the estimation of theoretical upper and lower bounds on AI and FAA to be consistent to the actual values. In other words, the actual values belong to a small range given by the estimated theoretical lower and upper bound (using Algorithm 3). In particular, Algorithm 3 may return an exact theoretical value, if the decomposition of these functions always occur so that λ is too close to $n - 1 = 13$ or $n - 2 = 12$, where $\lambda = \min\{i \mid ||B_i|| \neq 0, i = (1, \ldots, n-1)\}$. (In this case, it usually means that a Boolean function has quite good algebraic properties). For instance, in Example 5.2.4, we could easily verify that the actual resistance of this function against AA is $AI(f) = 7$. Moreover, using Algorithm 3 to estimate the theoretical lower and upper bound on AA, we found $r'+d = 0+8 = 8$, $r+d = 1+8 = 9$ and $\triangle_{AA}^{lower} = \nabla_{AA}^{upper} = 7$ which also completely consistent to the actual value $AI(f) = 7$.*

Example 5.2.6 *We select an $n = 10$ variable Boolean function $f(x)$ with good algebraic properties whose truth table in the hexadecimal format is given below. Using algorithms in [3], we could easily verify the actual resistance of this function against AA and FAA to be $AI(f) = 5$, $deg(g)+deg(h) \geq 9$, ($deg(f) = 8$) for nonzero Boolean functions g and h such that $fg = h$. In this case, $f(x)$ has an optimal AI and a suboptimal resistance against FAA. One decomposition of this function gives $||B_7|| = 3, ||B_8|| = 126, ||B_9|| = 248, ||B_i|| = 0, i \neq (7, 8, 9)$, if the order of fixing the input variables is $(x_4, x_7, x_1, x_6, x_9, x_8, x_3, x_5, x_2, x_{10})$. Using Algorithm 3 to estimate the theoretical lower and upper bound on AA, we get $AI(f) = 5$, ($\triangle_{AA}^{lower} = r' + d = 0 + 5 = 5, \nabla_{AA}^{upper} = 5$), which is consistent to the exact value $AI(f) = 5$. Moreover, another decomposition of this function is given by $||B_7|| = 9, ||B_8|| = 108, ||B_9|| = 260, ||B_i|| = 0, i \neq (7, 8, 9)$, if the order of fixing the input variables is $(x_8, x_4, x_7, x_9, x_6, x_3, x_2, x_5, x_1, x_{10})$. Using Algorithm 3 to estimate the theoretical lower and upper bound on the ability against FAA, we obtain $deg(g)+deg(h) \geq 9$, ($\triangle_{FAA}^{lower} = r'+s+e = 0+4+5 = 9, \nabla_{FAA}^{upper} r+s+e = 9$), which is also consistent to the actual value $deg(g)+deg(h) \geq 9$.*
1bdd 12ea 02eb a024 d67d d7e3 6a3c e80e 0fb8 c099 9fbd cc9c e961 3e2b 1803 2d93 5ed1 564e 5225 558e c9a5 e528 c022 56fd 1e93 f714 85cb fe18 2fbb 5241 a70a 3b5e 741b 13bf b36e d16f c83c 2e10 f06a 74a2 551c 3843 2768 959a c265 49d1 cfcb be8b 71dc b58d 0602 e8f2 5ee7 f048 61d9 76e5 a253 f153 ca70 caed 33c2 6027 4f5e 8c36

Example 5.2.7 *We use Algorithm 3 to check the theoretical upper bound on the resistance of functions in [124] against AA and FAA (note that some similar func-*

tions are also proposed in [125, 126]), where $||B_{\frac{n}{2}}|| = 2^{\frac{n}{2}-1}$, $||B_{\frac{n}{2}+1}|| = 2^{\frac{n}{2}-1}$, $||B_{\frac{n}{2}+2}|| = 2^{\frac{n}{2}}$, *(for even* $n = 12$ *to* 40*). In Table 5.3 and Table 5.4 we compare the upper bounds on AA and FAA, respectively, for this class of functions to their optimal values. It is clear that the resistance of functions designed in [124] against AA and FAA are not optimal or suboptimal, for even* $n = 18$ *to* 40*.*

Table 5.3: Estimation the upper bound on the AI values of functions in [124].

n	r	d	$r+d$	Optimal
12	1	5	6	6
14	1	6	7	7
16	1	7	8	8
18	1	7	8	9
20	1	8	9	10
22	1	9	10	11
24	1	9	10	12
26	1	10	11	13
28	1	11	12	14
30	1	11	12	15
32	1	12	13	16
34	1	13	14	17
36	1	13	14	18
38	1	14	15	19
40	1	15	16	20

Table 5.4: Estimation the upper bound on the resistance of functions in [124] against FAA.

n	r	s	e	$r+s+e$	Suboptimal
12	1	3	6	10	11
14	1	4	7	12	13
16	1	4	8	13	15
18	1	5	8	14	17
20	1	5	9	15	19
22	1	6	10	17	21
24	1	6	10	17	23
26	1	7	11	19	25
28	1	7	12	20	27
30	1	8	12	21	29
32	1	8	13	22	31
34	1	9	14	24	33
36	1	9	14	24	35
38	1	10	15	26	37
40	1	10	16	27	39

Appendix

The truth table is described in the hexadecimal format, in particular, the most significant bit is the leftmost bit, e.g. (0001) = 1, etc.
7781 42ae f22d 8aec fd3e 8b57 802c 88c9 ce89 297b 4cce d599 bd82 922b 55fc 3a16 f30a 55f1 4eb7 a053 2e6a fe64 efc8 6ebc c48c b22b 1485 7433 3273 d922 8bae 7489 9b5f f561 56a9 3b3e c55a c06e 2065 d239 d1e3 a264 a2b6 7fe2 a678 950e 008f 0695 92ef d039 0717 1fc6 71aa b196 8995 7e4f 8ca5 e200 d4a5 b60f 63f4 eb32 2a74 4cb0 ee6c d30e 3078 a31c c25c 5830 91f1 1ebc 8cf1 eb9d cb91 249f 4d25 917c 8572 e2bb 296e d5df 8dd1 81a8 c235 0e64 21d8 872a b366 49a6 fbbc 18c5 5cbf 71cb bdaa d167 a080 c782 0bad b799 1a25 b4bb 4735 0698 fe72 9ab7 312c 1390 890c 40a4 9344 8855 c4c1 c5b7 bb84 f631 abbe 6b80 fa0b 6c9b f01b af4a aca8 721e 95c7 0231 58ef ecf7 04ee d85d a353 049d 4b08 db7e 3fef d10c 1844 edb5 554e 5e97 b48e ab12 132f 698a df59 861c 4f86 8020 b72b 3006 8191 3e44 e0b2 7653 f7af 9f25 d973 888f 78b2 355b 9f0c 0a9b 2fac e83f b2ef 02e5 f309 dc3f 9bc1 df2f c573 4d59 203b 5c16 81b0 1ec5 e0c2 1ccd 8304 79cc 37a3 8c55 61ff c490 3fdf 9913 7e29 8657 c8b0 3f38 6530 0812 37e9 9e58 9877 ba62 28ad ea76 5601 f73e cc7e 841b 3997 9bc7 f825 bc1b a239 db33 3790 3b0b 5450 2586 e031 53fc db34 061b 8721 0b8b 8d35 f4b6 07bc e86c a023 b203 0dfd 2106 122f de79 d841 e718 fafc a8ae 60f1 888f aa40 e68e 3062 faaa 399b fd11 a816 b4f4 9bef 69da 7bca 74f5 f94e 5566 d381 77c3 f922 3d06 b68a 4ddf 2b13 f1ca 1920 3efb 5a83 3016 9ceb 3a77 a0f6 8c53 d371 fcab 704c ce36 91c9 bc18 3f15 7107 a27e 7ec9 7772 9549 1671 4268 67ed f431 bb6f e79f 36fd 0d31 0f0f 6c21 ded9 1a9d 71b3 4eb7 ba37 0c06 5a25 aac2 a8ac ce09 b8e1 c023 7283 46de 5361 4d8a 6e7c a514 0382 5f77 bee7 b05c 0bf6 68e8 8f26 1544 c6aa 6125 6a0e c458 7f6b 4b41 188c 0257 1626 6345 71a6 0ffc 2209 8d8a 59e7 1219 328f e78b 543d e9a5 c2c8 5ad6 d44e 9551 97e1 c67d 9a78 4efb 8de1 e1ae 23fa 5967 1f0f 6803 60e3 ae27 1a7e 5f51 d6e2 e9f9 04d5 39dd c4f0 93ed 8dc5 940e 5b8e 6f15 0023 a091 aedb 0469 91b7 a86a e9a2 6e25 8208 b40a 89fa 7fb5 bb50 6618 d243 f387 5577 f083 0cd8 b4cc 802f aa5c d930 bf3c 5a99 c06c 8dea 29a6 0fe8 5ec8 ffd8 7f18 a99e 30cd d5a4 c0b0 d8cd e626 7138 c026 b6f0 4217 2b09 c37c 7007 8008 fda5 7f98 9439 6ff4 109f 4878 1918 f9bd faea 6933 d666 cd06 6b9c 75d3 ac81 f138 1edb 9af2 cec6 a84f 3734 d2bf 13d0 c475 c1c0 a266 9755 cfa8 7adf af7e 84cf 7419 3261 1583 5a33 13d3 d501 08a6 3038 1a78 d253 0c84 596a 2bf6 18ca db4d 2590 5a1b 32c5 0ffd e21f 9129 a18c 6930 8bbf a26e 70d4 3880 994d ad0f 222c 2a96 c6b0 28a3 c424 e694 7f03 beef 108b 370b 7c02 1f49 b18f ee33 8b8e 92ca bdfe 35de 352a 7853 21c8 5788 1a4a 9b9f 3fec 0cb8 a8aa 7457 c3cf b66f f9bd 5545 0cc7 e8d3 9f6b 02f5 d92d 4b35 9588 0080 cd38 fe53 23d4 ac46 2ff3 9b3e a218 bf1b 2b85 6fba 3000 c683 6682 4fdd f861 21d5 9967 c98a f8d0 4b7d 0cb5 bbef 23d7 88d1 1652 6cc9 2712 0189 e538 1667 0e33 684e 8872 029d 474c efd1 e884 ebf8 357e b193 e41c 17da 0810 6907 9ca0 8d2f 73c5 832f 3088 f062 ca22 947e b220 8219 a4bf 908f b40d 0c1a e187 7372 8404 6394 7794 8190 e57b bafa 3d34 62f2 830c 617e 2bf3 de54 8a52 d89d 212a 2167 95b2 4f51 aeb9 22a7 9afa 86fb 4b9d df88 64e0 8da0 ebda fff9 60ed 518c b477 cd5a 9226 7b8f 6b9c 565a 619b 41c8 0c45 3c28 c774 34d1 1872 5d73 9027 fcad 59b6 46f2 ec07 75e4 3a57 27bb 12a9 f365 d6fe 4e43 8f51 340e 4bb4 b5b8 95c4 8d66 7723 2b5a 8fb4 43ba 96f2 51a7 a694 a037 efb2 b226 6146 8ccd e566 4b93 0d77 86a2 ebf0 aec6 8d36 3f3e 04c9 3d78 c809 cb35 bee8 b430 846e d2d8 f71e 6544 02aa 6292 28b7 3375 2f0e 8d39 0494 8162 1e47 8ac6 a838 d36f 60d0 991a

e8b0 5df4 3c66 26c7 dde2 d730 a238 a79e 9272 43dd e834 d173 bf99 8334 6c23 6080
2bfc 1394 d8a3 c6c2 37c0 f841 24ee 3b6e 30b0 b26f ed0a 50c2 645b 5bed 18cc 69da
fd96 1278 ce7d 04af d6ac ec0c e916 b9d4 8b43 4d29 1f44 5ea9 5e1c af72 2882 cee4
2405 8681 29a8 d263 f3a6 cd74 d8ef 18bf eeca e074 3338 e1ab 6beb 6d96 6583 7ab9
027e 5c4a 3548 0810 632f 8890 de52 aa75 6c7c ecbe 3b9d 6313 6f56 6cad f00d 5a12
cefe 25c7 49de cad6 1149 3f71 4474 4fde e1b2 e454 0fd5 72ff ac52 0dfd a431 f830
67f7 acc8 7f57 8a2c a5e4 d246 efc6 dea6 e811 6010 1358 07c9 d60e 8705 a59f bb44
036d 8976 c1be 5764 7150 a12e 1e38 ae78 ba6c acae be65 ceaf d3b9 ec8b ebea f99a
2b4a e777 dd69 cf46 5c39 8364 e18d 2072 e2b3 a147 122d 7f18 fb35 5e34 b95e 6249
8554 a956 e2fc 7950 8c40 734a 1be7 1bbb d8da 93d3 b127 0e2f acf8 1279 6572 7e7b
4cb4 ef99 dad8 547c 15f9 7329 6a89 31ca 0d74 1e41 1aa0 5fc8 21fd 1759 626e a379
fc08 35f8 374e 1520 a3d6 f8e6 c6d6 67eb a280 4702 b832 b3e6 9a87 fdec 0f28 ce0d
5975 7789 8a2b 71f9 b2c5 d04a 533d 014c 0872 6d1f c7cf e94d 1938 e4b7 e55b 8096
d03f 9878 7046 0138 9237 d1f1 2112 0073 e208 4c28 1b24 62ac a506 db48 6806 69b2
d9c1 df7f d4c5 463a 064c 8952 99e5 2277 42e6 78cf f304 4c75 2f7e acf3 8a6b c688
dbcf 96be 0eed ec01 2168 eee3 340d 0134 0f3f 7ad2 4dd0 0496 a347 7e99 8769 6310
7edb 7b56 b99b 43e2 8691 eeb4 923f 4d7e 0bf4 1bc6 8ef2 f74d 6f4d c730 b9ff 2553
d224 87a0 db1c 1e1c b774 5f25 fe70 9f99 cbb8 49fc b34b 9e4c af01 73a8 dd68 dbd4
5214 5b58 0724 9416 4e3f 08d5 7d1f d3da 84c7 24b2 d7bc 019a 2dc5 ca18 623b 67a6
df7f c59a e9c9 f230 a971 587a a07a a8ba 7229 a793 10ef c7aa 549a c6cd 311a d4af
a40e 99cb c87b 8522 6106 435d bd9e 8cae c26e e078 15bc 5d31 194d f731 9eb8 12bd
c597 7d42 8c24 6e1d 9043 880c d49c ac85 33c9 c489 a7b6 cf6b cf11 8996 5021 16fd

Chapter 6

On derivatives of polynomials over finite fields through integration

For a given polynomial $F(x) \in \mathbb{F}_q[x]$ its derivative at $a \in \mathbb{F}_q^*$ is defined as $D_aF(x) = F(x+a) - F(x)$, where clearly $a = 0$ results in a trivial annihilation. In contrast to the standard notion of derivative, which is for instance useful for determination of multiple roots of F and which coincides to the derivation of polynomials over real numbers, this notion of derivatives is of great importance in cryptography and is directly related to differential properties of the mappings used in the substitution boxes. Indeed, when $p = 2$ the differential properties of F (that reflects the resistance to differential cryptanalysis [4]) are characterized by the number of solutions of $F(x+a) + F(x) = b$ for any $a \in \mathbb{F}_q^*$ and $b \in \mathbb{F}_q$. On the other hand, for fields of odd prime characteristic $p > 2$, if $F(x+a) - F(x)$ is a permutation for any nonzero a then F is called a planar function [28, 21, 22]. The concept of linear structures plays an important role in cryptographic applications. Recall that for a polynomial $F(x) \in \mathbb{F}_{2^n}[x]$, represented as $F(x) = \sum_{i=0}^{q-1} b_i x^i$, an element $a \in \mathbb{F}_{2^n}$ is called b-linear structure ($b \in \mathbb{F}_{2^n}$) if the equality $F(x+a)+F(x) = b$ holds for every $x \in \mathbb{F}_{2^n}$. A few general results are known about the form of polynomials $F(x)$ admitting linear structures [19, 20, 121, 114]. The same applies to the Boolean case when $f : \mathbb{F}_{2^n} \to \mathbb{F}_2$ which again may be represented as $f(x) = \sum_{i=0}^{2^n-1} a_i x^i$ but the coefficients a_i must satisfy certain conditions, see relation (2.2). In [121], the properties of the set of differential functions defined as $\mathcal{DF}_q = \{D_aF(x) : F(x) \in F_q[x], a \in \mathbb{F}_q^*\}$ was investigated. One should notice that there exist polynomials in $\mathbb{F}_q[x]$ which are not derivatives of any polynomial, thus they do not belong to $\mathcal{DF}_q$. The main result in [121] concerning the existence of linear structures is that $F(x) \in \mathbb{F}_{2^n}[x]$ is a differential function (thus $F(x) \in \mathcal{DF}_q$) if and only if it has a 0-linear structure. This implies that the necessary condition to avoid linear structures is that $F(x) \notin \mathcal{DF}_q$, for $q = 2^n$. In [19], the authors investigated the existence of linear structures for the mappings of the form $F(x) = Tr(\delta x^s)$, where $F : \mathbb{F}_{p^n} \to \mathbb{F}_p$. For polynomials over finite fields a thorough treatment of binomials $F(x) = x^s + \alpha x^d$ was taken in [20].

The case of the discrete integration in finite fields of characteristic two and some result on the 0-linear structures of higher-order derivatives were studied recently in [114].

In this chapter we firstly derive the relationship between the coefficients b_i of $F(x) = \sum_{i=0}^{q-1} b_i x^i$ and the coefficients c_i of its derivative $G(x) = F(x+a) - F(x) = \sum_{i=0}^{q-2} c_i x^i$ (Section 6.1). This connection can be efficiently used for specifying conditions regarding the existence of linear structures for either Boolean functions or for mappings over finite fields. Though the approach is quite elementary it leads to several important results in this direction. For instance, it is sufficient that $F(x)$ contains the highest polynomial degree term x^{q-1} so that F does not admit linear structures, which when translated into the domain of Boolean functions corresponds to a class of functions of highest algebraic degree. Noticing that any n-variable Boolean function can also be represented as a univariate polynomial $f(x) = \sum_{i=0}^{q-1} b_i x^i \in \mathbb{F}_{2^n}[x]$, where the coefficients b_i satisfy certain conditions, we apply the same technique to either mappings over finite fields or to Boolean mappings. While the linear structures of monomials and binomials are quite easy to handle, in general the existence of linear structures for arbitrary polynomials is harder to analyze. Nevertheless, we provide a few interesting results in this direction covering also some particular cases when F contains an arbitrary number of terms (Section 6.1.2). Finally, using the same technique we provide a nontrivial upper bound on the degree of planar mappings (Section 6.2). Results of this chapter are published in [89].

6.1 Linear structures and derivatives

Throughout this chapter we write $F(x) = \sum_{i=0}^{q-1} b_i x^i$ and $D_{F,a}(x) = F(x+a) - F(x) = G(x) = \sum_{i=0}^{q-2} c_i x^i$, where $b_i, c_i \in \mathbb{F}_q$ and $a \in \mathbb{F}_q^*$, for $q = p^n$. Thus, given $D_{F,a}(x)$ specified by the known coefficients c_i our goal is to recover the values of b_i (or possibly a set of different polynomials $\{F\}$) so that the derivative of F at a corresponds to $G(x)$. For convenience, we sometimes write,

$$F(x) = \sum_{i=0}^{q-1} b_i x^i = \sum_{\substack{i=1 \\ i \neq p^j; 0 \leq j \leq n-1}}^{q-1} b_i x^i + \big(b_0 + \sum_{j=0}^{n-1} b_{p^j} x^{p^j}\big) = F^*(x) + A(x), \qquad (6.1)$$

where $A(x) = b_0 + \sum_{j=0}^{n-1} b_{p^j} x^{p^j}$ denotes an affine polynomial in $\mathbb{F}_q[x]$. Also, $A(x) = b_0 + L(x)$, where L is a linearized polynomial. Furthermore, denote by $\mathcal{L}_q$ and $\mathcal{A}_q$ the sets of all linearized and affine polynomials over $\mathbb{F}_q$, respectively, where $q = p^n$. Since for any $G, H \in \mathbb{F}_q[x]$ we have $D_{G+H,a}(x) = D_{G,a}(x) + D_{H,a}(x)$, then $D_{F,a}(x) = D_{F^*,a}(x) + D_{A,a}(x) = D_{F^*,a}(x) + L(a)$ due to the fact that $D_{A,a}(x) = L(a)$.

In general, for a given $a \in \mathbb{F}_q^*$ and $G(x)$ the coefficients b_i such that

$$F(x+a) - F(x) = G(x) \qquad \text{for all } x \in \mathbb{F}_q,$$

can be easily derived. Namely, using

$$F(x+a) - F(x) = \sum_{i=0}^{q-1} b_i[(x+a)^i - x^i] =$$

$$\sum_{i=0}^{q-1} b_i \left[\sum_{t=0}^{i} \binom{i}{t} x^t a^{i-t} - x^i \right] = \sum_{i=0}^{q-1} b_i \left[\sum_{t=0}^{i-1} \binom{i}{t} a^{i-t} x^t \right] = \sum_{t=0}^{q-2} \left[\sum_{i=t+1}^{q-1} \binom{i}{t} a^{i-t} b_i \right] x^t,$$

the following equations relating a, b_i and c_t is valid

$$c_t = \sum_{i=t+1}^{q-1} \binom{i}{t} a^{i-t} b_i, \quad \text{for } t = 0, 1, \ldots, q-2. \tag{6.2}$$

The set of equations can be written as

$$\begin{array}{llll} \binom{1}{0}ab_1 + & \binom{2}{0}a^2b_2 + & \ldots + & \binom{q-1}{0}a^{q-1}b_{q-1} = c_0 \\ & \binom{2}{1}ab_2 + & \ldots + & \binom{q-1}{1}a^{q-2}b_{q-1} = c_1 \\ & & \ddots & \vdots \quad\quad \vdots \\ & & \binom{q-2}{q-3}ab_{q-2} + & \binom{q-1}{q-3}a^2b_{q-1} = c_{q-3} \\ & & & \binom{q-1}{q-2}ab_{q-1} = c_{q-2}. \end{array} \tag{6.3}$$

In particular, if $q = p$ then all the diagonal coefficients are of the form $\binom{k}{k-1}a = ka$, for $k = 1, 2, \ldots, p-1$, and since these are nonzero the system has a unique solution.

For $q = p^n$ and $n > 1$, we have $\binom{p^u}{t} \equiv 0$, for all $t \neq 0, p^u$. Furthermore, on the main diagonal we have the coefficients $\binom{k}{k-1}a = ka \equiv 0 \mod p$, for all $k = ps$, where $s = 0, 1, \ldots, \frac{q}{p} - 1$. The last p equations of the above system are of the form:

$$\begin{array}{llllll} \binom{q-p}{q-p-1}ab_{q-p} & + \binom{q-p+1}{q-p-1}a^2b_{q-p+1} & + \binom{q-p+2}{q-p-1}a^3b_{q-p+2} + & \ldots & + \binom{q-1}{q-p-1}a^pb_{q-1} & = c_{q-p-1} \\ & \binom{q-p+1}{q-p}ab_{q-p+1} & + \binom{q-p+2}{q-p}a^2b_{q-p+2} + & \ldots & + \binom{q-1}{q-p}a^{p-1}b_{q-1} & = c_{q-p} \\ & & \ddots & & \vdots & \vdots \\ & & & \binom{q-2}{q-3}ab_{q-2} & + \binom{q-1}{q-3}a^2b_{q-1} & = c_{q-3} \\ & & & & \binom{q-1}{q-2}ab_{q-1} & = c_{q-2} \end{array}$$

The last $p-1$ equations can be uniquely solved for $b_{q-1}, \ldots, b_{q-p+1}$ recursively, but the first equation has to be a linear combination of the last $p-1$ equations, as $\binom{q-p}{q-p-1} \equiv 0 \pmod p$. Therefore, the coefficient c_{q-p-1} depends on a and on the coefficients $c_{q-2}, \ldots, c_{q-p}$ and furthermore b_{q-p} is free due to the fact that $\binom{q-p}{q-p-1} \equiv 0 \pmod p$. This also implies that the derivative $G(x)$ cannot be arbitrary due to this restriction on c_{q-p-1}. Similarly, by considering the last $2p$ equations of the system, the fact that $\binom{q-2p}{q-2p-1} \equiv 0$ implies that b_{q-2p} is free. Since the diagonal coefficient with b_{q-p} is zero, we can choose b_{q-p} to be arbitrary but fixed and evaluate uniquely the coefficients $b_{q-p-1} \ldots, b_{q-2p+1}$, but again c_{q-2p-1} will depend on a and on $c_{q-2}, \ldots, c_{q-2p}$. The same reasoning applies if we take p more equations.

In general, on the diagonal we have $\binom{sp}{sp-1} \equiv 0$, for $s = 0, 1, \ldots, \frac{q}{p} - 1$, and thus the coefficients b_{sp} are free (can be chosen arbitrary) but the corresponding equations are linear combinations of the equations below so the coefficient c_{sp-1} is not arbitrary but it is determined by this linear combination, i.e., with a and c_k

where $k > sp - 1$. Note that the system has q/p free coefficients and therefore $q^{(q/p)}$ distinct solutions $F(x)$. On the other hand, given arbitrary $G(x)$ there may not exist any function $F(x)$ such that $G(x)$ is its derivative for some $a \in \mathbb{F}_q$. The reason for this is that q/p coefficients in $G(x)$ are determined by other coefficients.

6.1.1 Some preliminary results using integration formula

It is of interest to investigate whether the differentiation of two polynomials whose difference is not an affine polynomial can give rise to same derivatives for the same/different values of a.

We notice that for a fixed $a \in \mathbb{F}_q^*$ the derivative $F(x+a) - F(x) = G(x)$ gave rise to a set of distinct functions $\{F\}$ whose cardinality is $q^{q/p}$, for $q = p^n$. On the other hand, for a given $F(x)$ the set $\mathcal{F}_A = \{F(x) + A(x) : A(x) \in \mathrm{A}_n\}$ is of cardinality q^{n+1} which is significantly smaller than $q^{p^{n-1}}$, for $n > 3$. This implies that there are other functions which are not in $\mathcal{F}_A$ whose derivative is G.

In the above analysis we have assumed that $a \in \mathbb{F}_q^*$ is fixed, but if this is not the case one can in general consider the problem of finding (non)distinct functions whose derivatives (taken at different values $a \neq a' \in \mathbb{F}_q^*$) are the same.

Proposition 6.1.1 *For a given function $F(x) = \sum_{i=0}^{q-1} b_i x^i$, $q = p^n$, such that $p \not| \deg(F)$, the condition $F(x+a) - F(x) = F(x+a') - F(x)$ implies $a = a'$, unless $b_i = 0$ for all $i \not\equiv 0 \pmod p$.*

Proof: Assume $\deg(F) = m$. The largest nonzero coefficient of both $D_{F,a}(x)$ and $D_{F,a'}(x)$ being c_{m-1}, we have

$$\binom{m}{m-1} a b_m = c_{m-1} = \binom{m}{m-1} a' b_m.$$

Since $\binom{m}{m-1} = m \not\equiv 0$ it immediately follows $a = a'$. Now assuming that $F(x+a) - F(x) = F(x+a') - F(x)$ for $a \neq a'$, then $b_m = 0$ and in general $b_i = 0$ for all $i \not\equiv 0 \pmod p$. ∎

Corollary 6.1.2 *If the field is of prime order then $F(x+a) - F(x) = F(x+a') - F(x)$, that is, $F(x+a) = F(x+a')$, implies $a = a'$.*

Notice that in the case $q = p$ the system (6.3) has a unique solution for any $a \in \mathbb{F}_p^*$, thus Corollary 6.1.2 implies that all the solutions are distinct.

Corollary 6.1.3 *Let $L(x)$ be a linearized polynomial over $\mathbb{F}_q$ such that $L(a) = L(a')$. Then $L(x+a) - L(x) = L(x+a') - L(x)$. In particular if $L(x)$ is a permutation over $\mathbb{F}_q$ then $L(x+a) - L(x) = L(x+a') - L(x)$ if and only of $a = a'$.*

Finally, we notice that taking $a \neq a'$ and considering (6.3) one may easily specify different functions F and F' such that $D_{F,a}(x) = D_{F',a'}(x)$.

In what follows we use a well-known result concerning the parity due to James W. L. Glaisher (also referred to as Luca's theorem).

Theorem 6.1.4 *Let n and k be two non-negative integers. Then,*

$$\binom{n}{k} \equiv \begin{cases} 0 \mod 2 & \text{if } n \text{ is even and } k \text{ is odd} \\ \binom{\lfloor n/2 \rfloor}{\lfloor k/2 \rfloor} \mod 2 & \text{otherwise.} \end{cases} \tag{6.4}$$

In general, using the base p expansions of $n = \sum_{i=0}^{r} n_i p^i$ and $k = \sum_{i=0}^{r} k_i p^i$, we have $\binom{n}{k} \equiv 0 \pmod{p}$ as soon as $n_i < k_i$ for at least one i, so that

$$\binom{n}{k} \not\equiv 0 \pmod{p} \text{ if and only if } k \preccurlyeq n,$$

where $k \preccurlyeq n$ means that $k_i \leq n_i$ for any $i = 0, \ldots, r$.

6.1.2 Linear structures of mappings over finite fields and Boolean functions

Obviously, the easiest way of applying the above result in the context of determining the existence of linear structures is to study sparse polynomials over finite fields. Notice that a linear structure $a \in \mathbb{F}_{2^n}$ of $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ means that $F(x+a)+F(x) = \gamma$ for all $x \in \mathbb{F}_{2^n}$ and some constant element $\gamma \in \mathbb{F}_{2^n}$. Furthermore, using the above notation, it is equivalent to saying that $c_i = 0$ for all $i \in [1, 2^n - 1]$ and $c_0 = \gamma$. In the case of monomials of the form $F(x) = b_r x^r$ we have the following result.

Theorem 6.1.5 *Let $F(x) = b_r x^r$ be a non-zero monomial, where $F(x) \in \mathbb{F}_{2^n}[x]$ and $1 \leq r \leq 2^n - 1$. Then, a is a non-zero linear structure of F if and only if $r = 2^i$ for some $i \in [0, n-1]$.*

Proof: If $r = 2^i$ so that $F(x) = b_{2^i} x^{2^i}$, then $F(x + a) + F(x) = b_{2^i} a^{2^i}$. Thus, any a is a linear structure of F. Conversely, assume that a is a linear structure of $F(x) = b_r x^r$ and consider

$$c_{r-k} = \binom{r-k+1}{r-k} a b_{r-k+1} + \ldots + \binom{r}{r-k} a^k b_r,$$

for some $1 \leq k \leq r$. Since b_r is the only nonzero b_i, we have $c_{r-k} = \binom{r}{r-k} a^k b_r$. Now if a is a linear structure, then $c_{r-k} = 0$ for all $k \in [1, r-1]$. Consequently, $\binom{r}{r-k} \equiv 0 \mod 2$ for these values of k. Especially, for $k = 1$ we have $\binom{r}{r-1} \equiv 0 \mod 2$ implying that r is even. Then, assuming $r > 2$, the condition that $\binom{r}{r-2} \equiv \binom{r/2}{r/2-1} \equiv 0 \mod 2$ (corresponding to c_{r-2}) implies that $r/2$ is even. Continuing this way, for any $k = 2^i$ we necessarily have that $r/2^i$ is even. Let $r = \sum_{j=0}^{n-1} r_j 2^j$ be the 2-adic representation of r and assume that v is the largest j such $r_j = 1$, thus $r_v = 1$ and $r_j = 0$ for $j > v$. Since k ranges from 1 to r, taking $k = 2^{v-1}$ implies that $r/2^{v-1}$ is also even. It means that $2^v \mid r$ and therefore r is of the form 2^v. ∎

A similar analysis can be performed for the case of binomials of the form $F(x) = x^d + ux^e$, but this has already been done in [20] where it was proved that $F(x)$ cannot have linear structures unless F is affine.

Remark 6.1.6 *For the Boolean case, when $p = 2$ and $F, G : \mathbb{F}_{2^n} \to \mathbb{F}_2$, the same reasoning as above applies though the coefficients of both F and G must satisfy the Boolean conditions mentioned in the introduction.*

In what follows, we derive some interesting results regarding the polynomial form of $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ regarding linear structures, where in the remainder of this section $F(x) = \sum_{i=0}^{2^n-1} b_i x^i$ and F either satisfies the Boolean conditions or not.

It is well-known that the presence of the highest degree term in the ANF of F, corresponding to the term x^{2^n-1}, implies unbalancedness of F (the converse is of course not true). This means that specifying $b_{2^n-1} = 1$ the function F is unbalanced and we show that in this particular case any such F cannot have linear structures. Assuming that a is a nonzero linear structure of F satisfying $b_{2^n-1} = 1$, then $c_{2^n-2} = 0$ and (6.2) gives for $t = 2^n - 2$,

$$c_{2^n-2} = \binom{q-1}{q-2} a b_{q-1} = a \cdot 1 = 0,$$

which then implies $a = 0$, a contradiction.

Theorem 6.1.7 *Let $F(x) = \sum_{i=0}^{q-1} b_i x^i$, $q = 2^n$, where $F : \mathbb{F}_2^n \to \mathbb{F}_2$ so that the coefficients of F satisfy the Boolean conditions. If $b_{q-1} = 1$ so that F is necessarily unbalanced, since its ANF contains the term $x_1 x_2 \cdots x_n$, then any such F does not admit linear structures.*

The importance of this result lies in the fact that any balanced Boolean function with good cryptographic properties apart from possibly having linear structures can easily be transformed into (just slightly) unbalanced function which does not possess linear structures. Moreover, the algebraic degree is then optimized.

Remark 6.1.8 *It is known that if a is all-one linear structure, that is $F(x + a) + F(x) = 1$, then F (which is Boolean) is necessarily balanced since the relation $F(x + a) = F(x) + 1$ means that F takes an equal number of ones and zeros. Nevertheless, the unbalancedness of F in Theorem 6.1.7, through the term x^{2^n-1}, also excludes all-zero linear structures.*

Let us proceed our investigation for the special case of potentially balanced functions F, thus requiring that $b_{q-1} = 0$. In this case, $a b_{q-1} = c_{q-2} = 0$ does not lead to a contradiction. Then, computing the next few relations between b_i and c_j from (6.2) (and constantly using $\binom{k}{k-1} = k \equiv 0 \mod 2$, for all $k = 2s$ where s is a positive integer) gives for $q = 2^n \geq 8$ the following

$$\begin{aligned}
c_{q-3} &= 0 = \binom{q-2}{q-3} a b_{q-2} + \binom{q-1}{q-3} a^2 b_{q-1} = a^2 b_{q-1} = a^2 \cdot 0 \\
c_{q-4} &= 0 = \binom{q-3}{q-4} a b_{q-3} + \binom{q-2}{q-4} a^2 b_{q-2} + \binom{q-1}{q-4} a^3 b_{q-1} \\
&= \binom{q-3}{q-4} a b_{q-3} + \binom{q-2}{q-4} a^2 b_{q-2}.
\end{aligned}$$

The first equation gives us no condition on b_{q-2}, it can be chosen arbitrary (since $\binom{q-2}{q-3} \equiv 0$) though if F is Boolean we must also have $b_{q/2-1}^2 = b_{q-2}$. The second equation depends on the parity of $\binom{q-3}{q-4}$ and $\binom{q-2}{q-4}$. Now, obviously $\binom{q-3}{q-4} = q-3 \equiv 1 \mod 2$, whereas $\binom{q-2}{q-4} \equiv \binom{q/2-1}{q/2-2} = q/2-1 \equiv 1 \mod 2$. This implies that the second equation above yields $b_{q-3} = ab_{q-2}$. In particular, since $b_i \in \mathbb{F}_2$ then assuming that either $b_{q-2} = 1$ or $b_{q-3} = 1$ we necessarily have that $a = 1$.

The expression for c_{q-5} given by

$$c_{q-5} = 0 = \binom{q-4}{q-5} ab_{q-4} + \binom{q-3}{q-5} a^2 b_{q-3} + \binom{q-2}{q-5} a^3 b_{q-2} + \binom{q-1}{q-5} a^4 b_{q-1},$$

requires again the analysis of the coefficients $\binom{q-3}{q-5}$ and $\binom{q-2}{q-5}$. Clearly $\binom{q-2}{q-5} \equiv 0 \mod 2$, since $q-2$ is even and $q-5$ is odd. Similarly, $\binom{q-3}{q-5} \equiv \binom{q/2-2}{q/2-3} = q/2-2 \equiv 0 \mod 2$. Thus, since also $\binom{q-4}{q-5} \equiv 0 \mod 2$, implies that b_{q-4} is arbitrary and at the same time $b_{q/2-2}^2 = b_{q-4}$. Similarly, computing

$$\begin{aligned} c_{q-6} &= 0 = \binom{q-5}{q-6} ab_{q-5} + \binom{q-4}{q-6} a^2 b_{q-4} + \binom{q-3}{q-6} a^3 b_{q-3} + \binom{q-2}{q-6} a^4 b_{q-2} \\ &\quad + \binom{q-1}{q-6} a^5 b_{q-1} = ab_{q-5} + a^4 b_{q-2}. \end{aligned}$$

implies that $b_{q-5} = a^3 b_{q-2}$ and also $b_{q-9} = (a^3 b_{q-2})^2$ using $b_{2i}^2 = b_i$.

Thus, in order to deduce stronger conditions on the coefficients we need to assume further restrictions on the form of F. Indeed, by requesting that $b_{q-2} = 0$ we necessarily have $b_{q-3} = b_{q-5} = 0$. Then, checking the expression for c_{q-7} which is given by,

$$\begin{aligned} c_{q-7} &= 0 = \binom{q-6}{q-7} ab_{q-6} + \binom{q-5}{q-7} a^2 b_{q-5} + \binom{q-4}{q-7} a^3 b_{q-4} + \binom{q-3}{q-7} a^4 b_{q-3} \\ &\quad + \binom{q-2}{q-7} a^5 b_{q-2} + \binom{q-1}{q-7} a^6 b_{q-1} = \binom{q-4}{q-7} a^3 b_{q-4} = a^3 b_{q-4}, \end{aligned}$$

taking into account that $b_{q-1} = b_{q-2} = b_{q-3} = b_{q-5} = 0$ and that $\binom{q-6}{q-7} \equiv 0 \mod 2$. But, since b_{q-4} is arbitrary then assuming it is non-zero leads to a contradiction $c_{q-7} = a^3 b_{q-4} \neq 0$. Therefore, assuming that $b_{q-1} = b_{q-2} = b_{q-3} = b_{q-5} = 0$ and $b_{q-4} \neq 0$ implies that such an $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ cannot have linear structures. Notice that the same reasoning is also valid for $F : \mathbb{F}_2^n \to \mathbb{F}_2$ since the Boolean conditions are actually irrelevant in the above derivation.

Theorem 6.1.9 *Let $F(x) = \sum_{i=0}^{q-1} b_i x^i$, $q = 2^n$, where $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$. If $b_{q-1} = b_{q-2} = b_{q-3} = b_{q-5} = 0$ and $b_{q-4} \neq 0$, then F cannot have linear structure. Furthermore, if the coefficients of F satisfy the Boolean conditions the same condition implies that $F : \mathbb{F}_{2^n} \to \mathbb{F}_2$ does not have linear structures.*

Remark 6.1.10 *The above result appears to be rather peculiar in the context of linear structures. There is no obvious reason why the above condition ensures the non-existence of linear structures. Certainly, there are other possibilities of specifying the coefficients b_i (for instance without forcing that $b_{q-2} = 0$) for the same purpose, though we do not explore this further.*

A similar analysis also implies the following result.

Theorem 6.1.11 *Let $F(x) = \sum_{i=0}^{2^n-1} b_i x^i$, $F \in \mathbb{F}_{2^n}[x]$, whose polynomial degree is d, where $d \in [1, 2^n - 1]$. Then,*

(i) If d is odd and $d > 1$, then F has no linear structures.

(ii) If d is even such that $4 \not| d$ and $b_{d-1} = 0$, then F has no linear structures.

(iii) If d is even such that $4 \mid d$ and $b_{d-1} = 1$, then F cannot have linear structures.

Proof: Throughout the proof we use the relation between $F(x)$ and its derivative $F(x) + F(x + a) = \sum_{i=0}^{2^n-2} c_i x^i$ given by (6.2).
(i) The case when $d = 2^n - 1$ follows from Theorem 6.1.7, regardless of whether b_{q-2} is zero or not. Thus, let $d < 2^n - 1$, where d is odd and $b_d \neq 0$. Since $c_i = 0$ for $i > d - 1$ let us consider

$$c_{d-1} = \binom{d}{d-1} a b_d = d a b_d \neq 0,$$

because $d \not\equiv 0$. Since $c_{d-1} \neq 0$ and $d - 1 > 0$, F does not have linear structures.
(ii) If $b_{d-1} = 0$ and d is even such that $4 \not| d$, then

$$c_{d-2} = \binom{d-1}{d-2} a b_{d-1} + \binom{d}{d-2} a^2 b_d = a^2 b_d \neq 0,$$

and F cannot have linear structures.
(iii) If $b_{d-1} = 1$ and d is even such that $4 \mid d$, then

$$c_{d-2} = \binom{d-1}{d-2} a b_{d-1} + \binom{d}{d-2} a^2 b_d = a b_{d-1} = a,$$

thus F cannot have linear structures in this case. ∎
Notice that the above result covers a large class of polynomials, having arbitrary number of terms, without linear structures. For instance, the main result in [20] was to establish the fact that binomials $F(x) = x^e + \alpha x^d$ cannot have linear structures unless F is affine. The result in Theorem 6.1.11 and a further simple analysis would lead to the same conclusion as already stated in [20].

6.2 Upper bounds on degree of planar mappings

In this section we will apply formulas for the integration of the polynomials to the planar mappings and consequently we deduce a nontrivial upper bound on the

polynomial degree of these mappings. Assume p is odd and that $F(x) = \sum_{i=0}^{q-1} b_i x^i$ is a planar polynomial, thus $p > 2$. Then, for all $a \in \mathbb{F}_q^*$, the polynomial $G(x) = F(x+a) - F(x) = \sum_{i=1}^{q-2} c_i x^i$ is a permutation, where the connection between the coefficients c_i and b_i has been established in the previous section.

Theorem 6.2.1 *Let $F(x) = \sum_{i=0}^{q-1} b_i x^i$ be a planar polynomial over $\mathbb{F}_q$, where the prime field of $\mathbb{F}_q$ is of odd characteristic and $q = p^m$ with $m \geq 2$ or if $q = p$, then $q \geq 7$. Then, the polynomial degree of F is less than or equal to $q - 1 - \frac{p+1}{2}$.*

Proof: For $F(x) = \sum_{i=0}^{q-1} b_i x^i$, if $G(x) = F(x+a) - F(x) = \sum_{i=1}^{q-2} c_i x^i$ is a permutation, then by Hermite's criterion $G^n(x) \pmod{x^q - x}$ has the coefficients with x^{q-1} equal to zero, for all $n = 1, 2, \ldots, q-2$. The case $n = 1$ implies $c_{q-1} = 0$.

Consider now $n = 2$. Squaring $G(x)$ we have that the coefficient d with x^{q-1} equals to $d = c_1 c_{q-2} + c_2 c_{q-3} + \cdots + c_{\frac{q-1}{2}}^2 + \cdots + c_{q-2} c_1 = \sum_{t=1}^{q-2} c_t c_{q-1-t}$. Using $c_t = \sum_{i=t+1}^{q-1} \binom{i}{t} a^{i-t} b_i$ and substituting in d we obtain

$$d = \sum_{t=1}^{q-2} \left(\sum_{i=t+1}^{q-1} b_i \binom{i}{t} a^{i-t} \right) \left(\sum_{j=q-t}^{q-1} b_j \binom{j}{q-1-t} a^{j-q+1+t} \right).$$

Let us use a new variable $s = i + j + 1 - q$. If $j = q - t$ then $s = i + 1 - t$ and for $j = q - 1$ we have that $s = i$. Therefore,

$$d = \sum_{t=1}^{q-2} \sum_{i=t+1}^{q-1} \sum_{s=i+1-t}^{i} \binom{i}{t} \binom{s+q-1-i}{q-1-t} b_i b_{s+q-1-i} a^s.$$

By changing the order of summation we obtain

$$d = \sum_{t=1}^{q-2} \sum_{s=2}^{q-1} \left(\sum_{i=\max\{s,t+1\}}^{\min\{s+t-1,q-1\}} \binom{i}{t} \binom{s+q-1-i}{q-1-t} b_i b_{s+q-1-i} \right) a^s =$$

$$\sum_{s=2}^{q-1} \left(\sum_{t=1}^{q-2} \sum_{i=\max\{s,t+1\}}^{\min\{s+t-1,q-1\}} \binom{i}{t} \binom{s+q-1-i}{q-1-t} b_i b_{s+q-1-i} \right) a^s.$$

We have that $d = 0$, for all $a \in \mathbb{F}_q^*$. Note that this is a polynomial in a, which is identically equal to zero for all $a \in \mathbb{F}_q^*$ and its degree is $q - 1$. Thus, all the coefficients with a^s, for $s = 2, 3, \ldots, q-1$, are equal to zero.

The coefficient with a^{q-1}, i.e., for $s = q - 1$, equals to

$$\sum_{t=1}^{q-2} \binom{q-1}{t} \binom{q-1}{q-1-t} b_{q-1}^2,$$

since $i = q - 1$. The binomial formula implies

$$\sum_{t=0}^{2(q-1)} \binom{2(q-1)}{t} y^t = (y+1)^{2(q-1)} = (y+1)^{q-1}(y+1)^{q-1} = \sum_{i=0}^{q-1} \binom{q-1}{i} y^i \cdot \sum_{j=0}^{q-1} \binom{q-1}{j} y^j.$$

Equating the coefficient with y^{q-1} we obtain the equality

$$\binom{2(q-1)}{q-1} = \sum_{i=0}^{q-1} \binom{q-1}{i}\binom{q-1}{q-1-i}.$$

Using this identity we obtain a simpler expression for the coefficient with a^{q-1} (note that the summation in the formula for the coefficient starts with 1)

$$\Big(\binom{2(q-1)}{q-1} - 2\Big)b_{q-1}^2 = \Big(\frac{2(q-1)\cdots q}{(q-1)!} - 2\Big)b_{q-1}^2 = -2b_{q-1}^2.$$

Since this coefficient is equal to zero we have $b_{q-1} = 0$.

Assume now that $b_{q-1} = \ldots = b_{q-u} = 0$, with $u < p/2$. Let us evaluate the coefficient with a^{q-1-2u}. Since $s = q-1-2u$, $\max\{q-1-2u, t+1\} = q-1-2u$, for $t \le q-2-2u$ and similarly $\max\{q-1-2u, t+1\} = t+1$, for $t > q-2-2u$. Also, $\min\{q-1-2u+t-1, q-1\} = q-1$ if $t \ge 2u+1$ and $\min\{q-1-2u+t-1, q-1\} = q-1-2u+t-1$, for $t < 2u+1$. If $q = p$ the only planar polynomials are quadratic so in this case theorem is satisfied. Assume $q = p^n$ where $n > 1$. Notice that $q > 2p+1$ for $u < p/2$ implies $q - 2u - 2 \ge 2u + 1$.
The coefficient with a^{q-1-2u} is

$$\sum_{t=1}^{q-2} \sum_{i=\max\{s,t+1\}}^{\min\{s+t-1,q-1\}} \binom{i}{t}\binom{s+q-1-i}{q-1-t} b_i b_{2(q-1)-2u-i} =$$

$$\sum_{t=1}^{2u} \sum_{i=q-1-2u}^{q-1-2u+t-1} \binom{i}{t}\binom{2(q-1)-2u-i}{q-1-t} b_i b_{2(q-1)-2u-i} +$$

$$\sum_{t=2u+1}^{q-2-2u} \sum_{i=q-1-2u}^{q-1} \binom{i}{t}\binom{2(q-1)-2u-i}{q-1-t} b_i b_{2(q-1)-2u-i} +$$

$$\sum_{t=q-1-2u}^{q-2} \sum_{i=t+1}^{q-1} \binom{i}{t}\binom{2(q-1)-2u-i}{q-1-t} b_i b_{2(q-1)-2u-i}.$$

Consider now the sum in the middle. If $i = q-1-2u, q-1-2u+1, \ldots, q-1-2u+(u-1) = q-2-u$ then $b_{2(q-1)-2u-i}$ equals to $b_{q-1} = b_{q-2} = \ldots = b_{q-u} = 0$. If $i = q-u, \ldots, q-1$ then $b_i = 0$ by assumption. For $i = q-1-u$ we have that $b_i b_{2(q-1)-2u-i} = b_{q-1-u}^2$. Therefore, the inner sum equals to

$$\sum_{t=2u+1}^{q-2-2u} \binom{q-1-u}{t}\binom{q-1-u}{q-1-t} b_{q-1-u}^2.$$

Consider now the first sum. Here, $q-2 \ge i \ge q-2-2u$. Similarly, the product $b_i b_{2(q-1)-2u-i} \ne 0$ only if $i = q-1-u \le q-1-2u+t-1$. There are nonzero terms only for $t \ge u+1$ and first sum equals to the

$$\sum_{t=u+1}^{2u} \binom{q-1-u}{t}\binom{q-1-u}{q-1-t} b_{q-1-u}^2.$$

Finally, let us consider the third sum. Here, i takes values $q-2u, q-2u+1, \ldots, q-1$. As already mentioned, $b_i b_{2(q-1)-2u-i} = 0$ for all values of i except for $i = q-1-u \geq t+1$ and thus $t \leq q-u-2$. Therefore, the third sum equals to

$$\sum_{t=q-1-2u}^{q-u-2} \binom{q-1-u}{t} \binom{q-1-u}{q-1-t} b_{q-1-u}^2.$$

The coefficient now is

$$\sum_{t=u+1}^{q-u-2} \binom{i}{t} \binom{q-1-u}{q-1-t} b_{q-1-u}^2 = b_{q-1-u}^2 \sum_{t=u+1}^{q-1-u-1} \binom{q-1-u}{t} \binom{q-1-u}{q-1-t}.$$

In order to simplify this expression consider

$$\sum_{t=0}^{2(q-1)-2u} \binom{2(q-1)-2u}{t} y^t = (y+1)^{2(q-1)-2u} = (y+1)^{q-1-u}(y+1)^{q-1-u} =$$

$$\sum_{i=0}^{q-1-u} \binom{q-1-u}{i} y^i \sum_{j=0}^{q-1-u} \binom{q-1-u}{j} y^j.$$

Equating the coefficient with y^{q-1} on both sides ($j = q-1-i$) we obtain equality

$$\sum_{i=u}^{q-1-u} \binom{q-1-u}{i} \binom{q-1-u}{q-1-i} = \binom{2(q-1)-2u}{q-1}.$$

Now we have that

$$\sum_{t=u+1}^{q-1-u-1} \binom{q-1-u}{t} \binom{q-1-u}{q-1-t} =$$

$$\binom{2(q-1)-2u}{q-1} - \binom{q-1-u}{u} \binom{q-1-u}{q-1-u} - \binom{q-1-u}{q-1-u} \binom{q-1-u}{u} =$$

$$\frac{(2(q-1)-2u)\cdots q \cdots (q-2u)}{(q-1)!} - 2\binom{q-1-u}{q-1-u} \binom{q-1-u}{u} \equiv -2\binom{q-1-u}{u} \pmod{p}.$$

Therefore, the coefficient is now

$$-2b_{q-1-u}^2 \binom{q-1-u}{u}.$$

Note that

$$\binom{q-1-u}{u} = \frac{(q-1-u)(q-1-u-1)\cdots(q-2u)}{u!} \neq 0 \pmod{p}$$

if $q-2u > q-p$, i.e., for $u < \frac{p}{2}$. Therefore, if $u < \frac{p}{2}$ we can conclude that $b_{q-1-u} = 0$. Inductively, we have that

$$b_{q-1} = b_{q-2} = \ldots = b_{q-\frac{p+1}{2}} = 0$$

for planar polynomials. ∎

If $q = p$ in the previous proof, successively considering $s = q - 1$, $s = q - 2$, we can show that $b_i = 0$ for all $i > \frac{q-1}{2}$. Applying the same idea to $(G(x))^3$, ..., $(G(x))^{p-2}$ it can be shown that the only planar polynomials over a prime field are quadratic, which is a well-known and established fact.

Corollary 6.2.2 *Assume that* $f(x) = \sum_{i=0}^{q-1} b_i x^i$ *is a planar polynomial. If there exists* $1 \leq s \leq n-1$ *where* $q = p^n$ *such that* $q - \frac{p+1}{2} \leq kp^s \mod (q-1) \leq q-1$ *then* $b_k = 0$.

Proof: If $f(x)$ is planar then $f(x^{p^s})$ is also planar where the coefficient with $x^{kp^s} \mod (x^q - x) = x^{kp^s \mod (q-1)}$ is b_k. Since the degree of f is less than $q - \frac{p+1}{2}$ we have that $b_k = 0$ if $kp^s \mod (q-1) \geq q - \frac{p+1}{2}$. ∎

Chapter 7

Conclusions

The results of the PhD Thesis represent a significant contribution to a number of the standing open problems in cryptography which have been an active topic of research in mathematical community in the last decades.

The major part of this thesis deals with the characterisation of generalized bent (gbent) functions (mappings from $\mathbb{Z}_2^n$ to $\mathbb{Z}_q$), where necessary and sufficient conditions are derived for q being a power of 2. Some of direct implications are determination of its dual function and the analysis of Gray maps. Additionally, the notion of $\mathbb{Z}_q$-bent functions is introduced and analyzed. Corresponding to $(2^n, 2^k, 2^n, 2^{n-k})$-relative difference sets in $\mathbb{F}_2^n \times \mathbb{Z}_{2^k}$, $\mathbb{Z}_q$-bent functions have its own interest in difference set theory. In difference to recent constructions of gbent functions for particular values of q, the first generic construction methods are also provided.

The importance of optimizing the placement of tap positions in LFSR-based ciphers lies in a fact that crypt-schemes (with tap positions which are inputs to filtering function) so far mainly used tap positions which correspond to difference sets, or tap positions which are selected heuristically. Namely, our analysis of GFSGA attacks and proposed algorithms shows that full positive difference sets (which are widely used) do not provide an optimal resistance to GFSGA-like attacks. We show that a significant improvement can be achieved, if our algorithms are used. Although that they do not provide an optimal selection of taps, which leaves space for further improvements, we actually show that the problem of taps selection requires more advanced methods which need to take in a consideration the nature of GFSGA-like attacks, and not only to rely on properties which come from difference sets.

Although several methods for estimating the resistance of a random Boolean function against (fast) algebraic attacks were proposed [25, 30, 3, 8, 29, 56], these methods are usually infeasible in practice for relative large number of input variables n (for instance $n \geq 30$) due to increased computational complexity. Introducing the concept of partial linear relations dissection, we develop an efficient probabilistic algorithm which estimates the resistance of Boolean function against (fast) algebraic attacks with time complexity about $O(n^2 2^n)$, thus offering much less complexity at the price of being probabilistic.

Using rather elementary techniques to connect the coefficients of a polynomial over a finite field and its derivatives, some new infinite classes of polynomials which

cannot possess linear structures are identified. The connection between the existence of linear structures and the differential profile of functions over finite fields is an important area of investigation in the context of the designs of S-boxes, since achieving the resistance against differential cryptanalysis is of a great importance. It is sufficient to mention that billion devices today are using Advanced Encryption Standard (AES), which is a block cipher designed upon S-boxes.

The basic tools used in the research range from combinatorial and algebraic methods in cryptography. An important tool in the study of gbent functions is the use of properties of cyclomatic fields and certain methods from linear algebra. An essential part in optimizing the placement of tap positions is detailed analysis of the GFSGA with a constant sampling rate. Since finding optimal solutions for tap selection is infeasible we rely on a sophisticated computer search using Mathematica software. To identify some infinite classes of polynomials which do not posses linear structures we rely on the theory of finite fields. Development of algorithms which estimate the resistance of Boolean function against AA and FAA is based on a novel method, which decomposes an arbitrary function into many small partial linear subfunctions by using the disjoint sets of input variables.

Bibliography

[1] M. Agren, M. Hell, T. Johansson, W. Meier. A new version of Grain-128 with optional authentication. *International Journal of Wireless and Mobile Computing*, vol. 5, no. 1, pp. 48–59, 2011.

[2] R. Anderson. Searching for the optimum correlation attack. In *Fast Software Encryption, FSE 94, Springe-Verlag*, vol. LNCS, pp. 137–143, 1995.

[3] F. Armknecht, C. Carlet, P. Gaborit, S. Knzli, W. Meier, O. Ruatta. Efficient computation of algebraic immunity for algebraic and fast algebraic attacks. *Advances in Cryptology–EUROCRYPT 2006* (Lecture Notes in Computer Science), Germany: Springer (Berlin), vol. 4004, 2006, pp. 147–164. 2006.

[4] E. Biham, A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[5] E. Biham, A. Shamir. Differential cryptanalysis of the Data Encryption Standard. In *Springer-Verlag, New York*, 1993.

[6] B. Bilgin, S. Nikova, V. Nikov, V. Rijmen, N. Tokareva, V. Vitkup Threshold implementations of small S-boxes. *Cryptography and Communications*, vol. 7, no. 1, pp. 3–33, 2015.

[7] A. Biryukov, A. Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In *Advances in Cryptology—ASIACRYPT 2000, Springer-Verlag*, vol. LNCS 1976, pp. 1–13, 2000.

[8] A. Braeken, J. Lano, B. Preneel. Evaluating the resistance of stream ciphers with linear feedback against fast algebraic attacks. *Information Security and Privacy* (Lecture Notes in Computer Science), Germany: Springer-Verlag (Berlin), vol. 4058, pp. 40–51, 2006.

[9] A. Braeken, B. Preneel. Probabilistic algebraic attacks. In *IMA Conference on Cryptography and Coding, Springer-Verlag*, vol. LNCS 3796, pp. 290–303, 2005.

[10] A. Braeken, J. Lano, N. Mentens, B. Preneel, and I. Verbauwhede. SFINKS: A synchronous stream cipher for restricted hardware environments. *eSTREAM, ECRYPT Stream Cipher Project*, Report 2005/026, 2005.

[11] C. D. Cannière, B. Preneel. Trivium: A stream cipher construction inspired by block cipher design principles. In *Information Security*, vol. LNCS 4176, pp. 171–186. Springer-Berlin Heidelberg, 2006.

[12] A. Canteaut, P. Charpin. Decomposing bent functions. *IEEE Transactions on Information Theory*, vol. 49, no. 8, pp. 2004–2019, 2003.

[13] C. Carlet, K. Feng. An Infinite Class of Balanced Functions with Optimal Algebraic Immunity, Good Immunity to Fast Algebraic Attacks and Good Nonlinearity. *Advances in Cryptology-ASIACRYPT 2008* (Lecture Notes in Computer Science), vol.5350. Berlin, Germany: International Association for Cryptologic Research, pp. 425–440, 2008.

[14] C. Carlet. Two new classes of bent functions. *Eurocrypt '93*, LNCS, vol. 765, pp. 77–101, 1994.

[15] C. Carlet. $\mathbb{Z}_{2^k}$-linear Codes. *IEEE Transactions on Information Theory*, vol. 44, no. 4, pp. 1543–1547, 1998.

[16] C. Carlet. On Bent and Highly Nonlinear Balanced/Resilient Functions and their Algebraic Immunities. *AAECC, Lecture Notes in Computer Science 3857, Springer-Verlag, New York*, pp. 1–28, 2006.

[17] C. Carlet. Boolean functions for cryptography and error correcting codes. *Cambridge University Press*, 2010.

[18] A. Çeşmelioğlu, W. Meidl, A. Pott. There are infinitely many bent functions for which the dual is not bent. *IEEE Transactions on Information Theory*, vol. 62, no. 9, 2016.

[19] P. Charpin, G. Kyureghyan. Monomial functions with linear structure and permutation polynomials. *Contemporary Mathematics*, vol. 518, pp. 99–111, 2010.

[20] P. Charpin, S. Sarkar. Polynomials with linear structure and Maiorana–McFarland construction. *IEEE Transactions on Information Theory*, IT-57(6), pp. 3796–3804, 2011.

[21] R. S. Coulter, R. W. Matthews. Planar functions and planes of Lenz-Barlotti class ii. *Designs, Codes and Cryptography*, vol. 10, pp. 167–184, 1997.

[22] R. S. Coulter, R. W. Matthews. Dembowski-Ostrom polynomials from Dickson polynomials. *Finite Fields and Their Applications*, vol. 15, no. 5, pp. 369–379, 2010.

[23] N. Courtois. Algebraic attacks on combiner with memory and several outputs. In *International Conference on Information Security and Cryptology – ICISC 2004, Springer-Verlag*, LNCS 3506, pp. 3–20, 2005.

[24] N. Courtois, W. Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology—EUROCRYPT 2003, Springer-Verlag*, LNCS 2656, pp. 346–359, 2003.

[25] N. T. Courtois, and W. Meier. Algebraic attacks on stream ciphers with linear feedback. *Advances in Cryptology–EUROCRYPT 2003* (Lecture Notes in Computer Science), Berlin, Germany: International Association for Cryptologic Research, vol. 2656, pp. 345–359, 2003.

[26] N. T. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. *Advances in Cryptology–CRYPTO 2003* (Lecture Notes in Computer Science), Germany: Springer-Verlag (Berlin), vol. 2729, pp. 176–194, 2003.

[27] D. Dalai, K. Gupta, S. Maitra. Notion of algebraic immunity and its evaluation related to fast algebraic attacks. *International Workshop on Boolean Functions: Cryptography and Applications*, pp. 13–15, 2006.

[28] P. Dembowski, T. G. Ostrom. Planes of order n with collineation groups of order n^2. *Mathematische Zeitschrift*, vol. 103, no. 3, pp. 239–258, 1968.

[29] F. Didier. Using Wiedemann's algorithm to compute the immunity against algebraic and fast algebraic attacks. *Progress in Cryptology–INDOCRYPT 2006* (Lecture Notes in Computer Science), Germany: Springer-Verlag (Berlin), vol. 4329, pp. 236–250, 2006.

[30] F. Didier, J. Tillich. Computing the algebraic immunity efficiently. *Fast Software Encryption* (Lecture Notes in Computer Science), Germany: Springer-Verlag (Berlin), vol. 4047, 2006, pp. 359–374.

[31] J. F. Dillon. Elementary Hadamard difference sets. *PhD Thesis*, University of Maryland, 1974.

[32] J. F. Dillon. Elementary Hadamard difference sets. *In proceedings of 6th S. E. Conference of Combinatorics, Graph Theory, and Computing, Utility Mathematics, Winnipeg*, pp. 65–76, 1975.

[33] H. Dobbertin. Construction of bent functions and balanced Boolean functions with high nonlinearity. *Fast Software Encryption, Springer-Verlag*, Leuven 1994, LNCS 1008, pp. 61–74, 1995.

[34] S. Dubuc. Characterization of linear structures. *Designs, Codes and Cryptography*, vol. 22, pp. 33–45, 2001.

[35] M. Engels. Wireless OFDM Systems: How to Make Them Work?. *The Springer International Series in Engineering and Computer Science, Springer US*, 2002.

[36] P. Ekdahl, T. Johansson. A new version of the stream cipher SNOW. *Selected Areas in Cryptography, International Workshop on Selected Areas in Cryptography, Springer Berlin*, LNCS 2595, pp. 47–61, 2002.

[37] J. H. Evertse. Linear structures in block ciphers. In *Advances in Cryptology–EUROCRYPT 1987, Springer-Verlag*, vol. LNCS 304, pp. 249–266, 1988.

[38] S. Fu, C. Li, L. Qu. A recursive construction of highly nonlinear resilient vectorial functions. *Information Sciences*, vol. 269, pp. 388–396, 2014.

[39] S. Gangopadhyay, E. Pasalic, P. Stănică. A note on generalized bent criteria for Boolean functions. *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 3233–3236, 2013.

[40] M. J. E. Golay. Complementary series. *IRE Transactions on Information Theory*, vol. 7, no. 2, pp. 82–87, 1961.

[41] J. Dj. Golić. Intrinsic statistical weakness of keystream generators. In *Advances in Cryptology—ASIACRYPT 1994, Springer-Verlag*, vol. LNCS 917, pp. 91–103, 1995.

[42] J. Dj. Golić. On the security of nonlinear filter generators. In *Fast Software Encryption'96, Springer-Verlag*, vol. LNCS 1039, pp. 173–188, 1996.

[43] J. Dj. Golić, Andrew Clark, Ed Dawson. Generalized inversion attack on nonlinear filter generators. *IEEE Transactions on Computers*, vol. 49, no. 10, pp. 1100–1109, 2000.

[44] L. Hanzo, C. H. Wong. Adaptive Wireless transceivers: Turbo-Coded, Turbo-Equalised and Space-Time Coded TDMA, CDMA, MC-CDMA and OFDM systems. *A John Wiley and Sons, Inc.*, 2002.

[45] L. Hanzo, T. Keller. OFDM and MC-CDMA: A Premier. *Wiley-IEEE Press*, 2006.

[46] P. Hawkes, G. Rose. Primitive specification and supporting documentation for SOBER-t16 submission to NESSIE. *In Proceedings of the First Open NESSIE Workshop, KU-Leuven*, 2000.

[47] M. Hell, T. Johansson, A. Maximov, W. Meier.. The Grain family of stream ciphers. *New Stream Cipher Designs, Springer Berlin*, LNCS 4986, pp. 179–190, 2008.

[48] M. Hellman. A cryptanalytic time-memory tradeoff. *IEEE Transactions on Information Theory*, vol. 26, no. 4, pp. 401–406, 1980.

[49] S. Hodžić, E. Pasalic, Y. Wei. Optimizing the placement of tap positions and guess and determine cryptanalysis cryptanalysis with variable sampling. Available at: https://arxiv.org/abs/1609.08422

[50] S. Hodžić, E. Pasalic. Generalized bent functions - Some general construction methods and related necessary and sufficient conditions. *Cryptography and Communications*, vol. 7, no. 4, pp. 469–483, 2015.

[51] S. Hodžić, W. Meidl, E. Pasalic. Full characterization of generalized bent functions as (semi)-bent spaces, their dual, and the Gray image. Available at: https://arxiv.org/abs/1605.05713

[52] S. Hodžić, E. Pasalic. Generalized bent functions - sufficient conditions and related constructions. Available at: http://arxiv.org/pdf/1601.08084v1.pdf.

[53] S. Hodžić, E. Pasalic. Construction methods for generalized bent functions. Available at: https://arxiv.org/abs/1604.02730

[54] J. Hong, P. Sarkar. New applications of time memory data tradeoffs. In *Advances in Cryptology–ASIACRYPT 2005, Springer-Verlag*, vol. LNCS 3788, pp. 353–372, 2005.

[55] G. Ivanov, N. Nikolov, S. Nikova Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties. *IACR Cryptology ePrint Archive*, 2014:801, 2014.

[56] L. Jiao, B. Zhang, M. Wang. Revised algorithms for computing algebraic immunity against algebraic and fast algebraic attacks. *Information Security* (Lecture Notes in Computer Science), Springer International (Switzerland), vol. 8783, pp. 104–119, 2014.

[57] L. Jiao, M. Wang, Y. Li, M. Liu. On annihilators in fewer variables basic theory and applications. *Chinese Journal of Electronics*, vol. 22, no. 3, pp. 489–494, 2013.

[58] L. R. Knudsen. Truncated and higher order differentials. *International Workshop on Fast Software Encryption, FSE 1994: Fast Software Encryption*, pp. 196–211, 1994.

[59] L. R. Knudsen. DEAL - A 128-bit block cipher . *Technical report no. 151. Department of Informatics, University of Bergen, Norway*, 1998.

[60] P. V. Kumar, R. A. Scholtz and L. R. Welch. Generalized bent functions and their properties. *Journal of Combinatorial Theory, Series A*, vol. 40, no. 1, pp. 90–107, 1985.

[61] P.V. Kumar, R.A. Scholtz, L.R. Welch. Generalized bent functions and their properties. *Journal of Combinatorial Theory, Series A*, vol. 40, no. 1, pp. 90–107, 1985.

[62] P. V. Kumar, R. A. Scholtz, L.R. Welch. Generalized bent functions and their properties. *Journal of Combinatorial Theory, Series A,* vol. 40, pp. 90–107, 1985.

[63] X. Lai. Higher order derivatives and differential cryptanalysis. *Communications and Cryptography, The Springer International Series in Engineering and Computer Science*, vol. 276, pp. 227–233, 1994.

[64] X. Lai. Additive and linear structures of cryptographic functions. In *Fast Software Encryption*, Second International Workshop, LNCS 1008, pp. 75–85, 1995.

[65] Y. Li, T. Cusick. Linear structures of symmetric functions over finite fields. In *Information Processing Letters*, vol. 97, no. 3, pp. 124–127, 2006.

[66] N. Li, X. Tang, T. Helleseth. New classes of generalized Boolean bent functions over $\mathbb{Z}_4$. *Proceedings of IEEE International Symposium on Information Theory ISIT 2012*, 2012.

[67] F. Liu, K. Feng. Perfect algebraic immune functions. *Advances in Cryptology - Asiacrypt 2012* (Lecture Notes in Computer Science), Germany: Springer-Verlag (Berlin), vol. 7658, pp. 172–189, 2012.

[68] H. Liu, K. Feng K, R. Feng. Nonexistence of generalized bent functions from $\mathbb{Z}_2^n$ to $\mathbb{Z}_m$. *Designs, Codes and Cryptography*, pp. 1–16, 2016. Available at: http://arxiv.org/pdf/1507.05751

[69] H. Liu, G. Li. OFDM-based broadband Wireless networks: Design and optimization. *A John Wiley and Sons, Inc.*, 2005.

[70] T. Martinsen, W. Meidl, P. Stanica. Generalized bent functions and their Gray images. Available at: http://arxiv.org/abs/1511.01438

[71] T. Martinsen, W. Meidl, P. Stanica. Partial spread and vectorial generalized bent functions. Available at: http://arxiv.org/abs/1511.01705

[72] T. Martinsen, W. Meidl, S. Mesnager, P. Stanica Decomposing generalized bent and hyperbent functions. Available at: https://arxiv.org/abs/1604.02830

[73] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.

[74] M. Matsui. Linear cryptanalysis method for DES cipher. *Advances in Cryptology, EUROCRYPT93, Springer-Verlag*, LNCS 765, pp. 386–397, 1993.

[75] M. Matsui. The First Experimental Cryptanalysis of the Data Encryption Standard. *Advances in Cryptology, EUROCRYPT94, Springer-Verlag*, LNCS 839, pp. 1–11, 1994.

[76] W. Meidl. A secondary construction of bent functions, octal gbent functions and their duals. *Mathematics and Computers in Simulation*, In Press, 2016.

[77] W. Meier, O. Staffelbach. Fast correlation attacks on certain stream ciphers. *Journal of Cryptology*, vol. 1, no. 3, pp. 159–176, 1989.

[78] W. Meier, E. Pasalic, C. Carlet. Algebraic attacks and decomposition of Boolean functions. In *Advances in Cryptology—EUROCRYPT 2004, Springer-Verlag*, vol. LNCS 3027, pp. 474–491, 2004.

[79] W. Meier, E. Pasalic, C. Carlet. Algebraic attacks and decomposition of Boolean functions. *Advances in Cryptology–EUROCRYPT 2004* (Lecture Notes in Computer Science), Germany: Springer-Verlag (Berlin), vol. 3027, pp. 474–491, 2004.

[80] S. Mesnager. Several infinite classes of bent functions and their duals. *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4397–4407, 2014.

[81] S. Mesnager. A Note on Linear Codes and Algebraic Immunity of Boolean Functions. *21st Int. Sym. Math. Theory Net. Sys.*(Groningen, the Netherlands), pp. 923–927, 2014.

[82] M. J. Mihaljević, M. P. C. Fossorier, H. Imai. A general formulation of algebraic and fast correlation attacks based on dedicated sample decimation. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, Springer-Verlag*, vol. LNCS 3857, pp. 203–212, 2006.

[83] M. J. Mihaljević, S. Gangopadhyay, G. Paul, H. Imai. Internal state recovery of Grain-v1 employing normality order of the filter function In *IET Information Security*, vol. 6, no. 2, pp. 55–64, 2006.

[84] K. Nyberg. Perfect nonlinear S-boxes. *EUROCRYPT'91 Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, pp. 378–386, 1992.

[85] K. Nyberg, L. R. Knudsen. Provable security against a differential attack. *Journal of Cryptology*, vol. 8, no. 1, pp. 27–37, 1995.

[86] M.G. Parker, A. Pott. On Boolean functions which are bent and negabent. *Sequences, subsequences, and consequences*, Lecture Notes in Computer Science, vol. 4893, pp. 9–23, 2007.

[87] E. Pasalic. Probabilistic versus deterministic algebraic cryptanalysis - a performance comparison. *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 2182–2191, 2009.

[88] E. Pasalic. On guess and determine cryptanalysis of LFSR-based stream ciphers. *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3398–3406, 2009.

[89] E. Pasalic, A. Muratović-Ribić, S. Gangopathyay, S. Hodžić. On derivatives of polynomials over finite field through integration. *Discrete Applied Mathematics*, vol. 217, no. 2, pp. 294–303, 2017.

[90] E. Pasalic, S. Hodžić, S. Bajrić, Y. Wei. Optimizing the Placement of Tap Positions. *Cryptography and Information Security in the Balkans—BalkanCryptSec 2014, Springer-Verlag*, LNCS 9024, pp. 15–30, 2015.

[91] A. Pott. Nonlinear functions in abelian groups and relative difference sets. *Discrete Applied Mathematics*, vol. 138, no. 1-2, pp. 177–193, 2004.

[92] A. Pott, K.U. Schmidt, Y. Zhou. Semifields, relative difference sets, and bent functions. *Algebraic curves and finite fields*, Radon Series on Computational and Applied Mathematics 16, De Gruyter, Berlin, pp. 161–178, 2014.

[93] A. Pott. Almost perfect and planar functions. *Designs, Codes and Cryptography*, vol. 78, no. 1, pp. 141–195, 2016.

[94] H. Rohling. OFDM: Concepts for future communication systems. *Signals and Communication Technology, Springer-Verlag Berlin*, 2011.

[95] P. Rogaway, D. Coppersmith. A software-optimized encryption algorithm. *Fast Software Encryption, International Workshop on Fast Software Encryption, Springer Berlin*, vol. 809, pp. 56–63, 1994.

[96] P. Rogaway, D. Coppersmith. A software-optimized encryption algorithm. *Journal of Cryptology*, vol. 11, no. 4, pp. 273–287, 1998.

[97] J. Robert, Jr. Jenkins. ISAAC. *Fast Software Encryption, International Workshop on Fast Software Encryption, Springer Berlin*, LNCS 1039, pp. 41–49, 1996.

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

[99] P. Sarkar, S. Maitra. Cross-correlation analysis of cryptographically useful Boolean functions and S-boxes. *Theory of Computing Systems*, vol. 35, no. 1, pp. 39–57, 2002.

[100] H. Schulze, C. Lueders. Theory and applications of OFDM and CDMA: Wideband Wireless communications. *A John Wiley and Sons, Inc.*, 2005.

[101] K. U. Schmidt. Quaternary constant-amplitude codes for multicode CDMA. *IEEE International Symposium on Information Theory, ISIT'2007*, Nice, France, June 2007. Available at http://arxiv.org/abs/cs.IT/0611162.

[102] K.U. Schmidt, Y. Zhou. Planar functions over fields of characteristic two. *Journal of Algebraic Combinatorics*, vol. 40, no. 2, pp. 503–526, 2014.

[103] K. U. Schmidt. $\mathbb{Z}_4$-valued quadratic forms and quaternary sequence families. *IEEE Transactions on Information Theory*, vol. 55, no. 12, pp. 5803–5810, 2009.

[104] K. U. Schmidt. Complementary sets, generalized Reed-Muller Codes, and power control for OFDM. *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 808–814, 2007.

[105] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal, Vol. 27:379423 (Part I) and 623656 (Part II)*, 1948.

[106] T. Siegenthaler. Decrypting a class of stream cipher using ciphertext only. *IEEE Transactions on Computers*, vol. C-34, no.1, pp.81–85, 1985.

[107] B. K. Singh. On cross-correlation spectrum of generalized bent functions in generalized Maiorana-McFarland class. *Information Sciences Letters*, vol. 2, no. 3, pp. 139–145, 2013.

[108] B. K. Singh. Secondary constructions on generalized bent functions. *IACR Cryptology ePrint Archive*, pp. 17–17, 2012.

[109] P. Solé, N. Tokareva. Connections between quaternary and binary bent functions. *IACR Cryptology ePrint Archive*, 2009. Available at https://eprint.iacr.org/2009/544.pdf

[110] V. I. Solodovnikov. Bent functions from a finite abelian group into a finite abelian group. *Discrete Mathematics and Applications*, vol. 12, no. 2, pp. 111–126, 2002.

[111] P. Stanica, T. Martinsen, S. Gangopadhyay, B. K. Singh. Bent and generalized bent Boolean functions. *Designs, Codes and Cryptography*, vol. 69, pp. 77–94, 2013.

[112] P. Stanica, S. Gangopadhyay, B. K. Singh. Some results concerning generalized bent functions, *IACR Cryptology ePrint Archive*, 2011. Available at https://eprint.iacr.org/2011/290.pdf

[113] P. Stanica, T. Martinsen. Octal bent generalized Boolean Functions. *IACR Cryptology ePrint Archive*, pp. 89–89, 2011.

[114] V. Suder. Antiderivative functions over $\mathbb{F}_{2^n}$. *Designs, Codes and Cryptography*, pp. 1–13, 2016.

[115] Y. Tan, A. Pott, T. Feng.. Strongly regular graphs associated with ternary bent functions. *Journal of Combinatorial Theory, Series A.*, vol. 117, no. 6, pp. 668–682, 2010.

[116] C. Tang, C. Xiang, Y. Qi, K. Feng. Complete characterization of generalized bent and 2^k-bent Boolean functions. *Cryptology ePrint Archive: Report 2016/335*, 2016.

[117] N. Tokareva. Bent functions: results and applications to cryptography. Academic Press – Elsevier, 2015.

[118] D. Wagner. The Boomerang attack. *International Workshop on Fast Software Encryption, FSE 1999: Fast Software Encryption*, pp. 156–170, 1999.

[119] Y. Wei, E. Pasalic, Y. Hu.. Guess and determinate attacks on filter generators–Revisited. *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2530–2539, 2012.

[120] Y. Wei, E. Pasalic, F. Zhang, S. Hodžić. Efficient probabilistic algorithm for estimating the algebraic properties of Boolean functions for large n. *Information Sciences*, vol. 402, pp. 91–104, 2017.

[121] H. Xiong, L. Qu, C. Li, Y. Li. Some results on the differential functions over finite fields. *Applicable Algebra in Engeneering, Communication and Computing*, vol. 25, no. 3, pp. 189–195, 2014.

[122] Y. Zhao, H. Li. On bent functions with some symmetric properties. *Discrete Applied Mathematics,* vol. 154, pp. 2537–2543, 2006.

[123] X. Zhang, B. Wu, Q. Jin, Z. Liu. Constructing Generalized Bent Functions from Trace Forms of Galois Rings. *Computer Mathematics - ASCM 2009*, pp. 467–477, 2014.

[124] W. Zhang, E. Pasalic. Generalized Maiorana-McFarland construction of resilient Boolean functions with high nonlinearity and good algebraic properties. *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 6681–6695, 2014.

[125] W. Zhang, G. Xiao. Constructions of almost optimal resilient Boolean functions on large even number of variables. *IEEE Transactions on Information Theory*, vol. 55, no. 12, pp. 5822–5831, 2009.

[126] W. Zhang, E. Pasalic. Improving the lower bound on the maximum nonlinearity of 1-resilient Boolean functions and designing functions satisfying all cryptographic criteria. *Information Sciences*, vol. 376, 21–30, 2017.

[127] F. Zhang, C. Carlet, Y. Hu, and T. Cao. Secondary Constructions of Highly Nonlinear Boolean Functions and Disjoint Spectra Plateaued Functions. *Information Sciences*, vol. 283, pp. 94–106, 2014.

[128] Y. Zhou. $(2^n, 2^n, 2^n, 1)$-relative difference sets and their representations. *Journal of Combinatorial Designs*, vol. 21, no. 12, pp. 563–584, 2013.

Index

$\mathbb{Z}_q$-bent, 16, 48

affine (semi-)bent space, 43
affine equivalence, 45
algebraic degree, 12
algebraic immunity, 107
algorithms, 83
annihilator, 107
associated algebraic normal form, 12
attack complexity, 19, 79

Boolean function, 12
  generalized dual, 50
  affine function, 12
  bent, 14
  dual, 15
  generalized, 15, 23
  generalized bent (gbent), 15, 31, 40
  plateaued function, 15
  semi-bent, 15
  vectorial, 12
  vectorial bent, 14

characters of the group, 14

derivative, 13
disjoint spectra, 14, 55
dot (scalar) product, 12

full positive difference sets, 96

Galois field, 11
GFSGA modes
  $GFSGA^*_{(1)}$, 93
  $GFSGA^*_{(2)}$, 94
Gray map, 51
guessing attack, 17
  filter-state (FSGA), 17
  generalized filter-state (GFSGA), 18

Hamming
  distance, 12
  weight, 12

lexicographic ordering, 11
linear structure, 127

Maiorana-McFarland class, 55

nonlinear filtering generator, 17
nonlinearity, 12

partial linear relations, 111, 113
planar mappings, 134
primitive element, 11

relative difference set, 14, 48

sequence of a function, 16
stream ciphers
  Grain-128, 103
  SFINX, 87
  SOBER-t32, 86
support of a function, 12
Sylvester-Hadamard matrix, 16, 38

tap positions, 18, 77
Trace function, 13
truth table, 12

univariate representation, 13

vector space, 11

Walsh-Hadamard transform, 13
  generalized, 15, 29
  normalized, 16

Chapter 8

Povzetek v slovenskem jeziku

KARAKTERIZACIJA POSPLOŠNIH ZLOMLJENIH FUNKCIJ IN NEKATERE DRUGE KRIPTOGRAFSKE TEME

Ljudje so že v antičnih časih želeli, da ostanejo določene, na papir zapisane, informacije zaupne. Družba še naprej zahteva metode za zaščito občutljivih informacij, a se je v informacijski dobi enkripcijska abeceda zreducirala na ničle in enke v elektronskih podatkih. Posledično je postal šifrirni proces vse bolj matematične narave. Tehnike za zaščito podatkov spadajo v kriptografijo, ki predstavlja znanost o informacijski in komunikacijski varnosti.

Poglavitni cilj kriptografije je omogočanje dvema osebama, da komunicirata preko nezaščitenega kanala na tak način, da nasprotnik (tretja oseba) ne more razbrati vsebine prvotnega sporočila (imenovanega čistopis) iz informacije, ki je bila poslana preko kanala (šifropis). Povedano splošneje, kriptografija sestavlja in analizira sisteme (protokole), ki onemogočajo branje zasebnih sporočil tretjim osebam. Na drugi strani kriptoanaliza preučuje kako ukaniti take sisteme. Obe vedi združuje kriptologija, ki preučuje komunikacijo preko nezaščitenih kanalov. Moderna kriptografija leži v preseku številnih ved v matematiki, računalništvu in elektrotehniki.

Uporabo kriptografije v družbi zasledimo v obliki avtentikacije, šifriranja (bančne kartice, brezžični telefoni, elektronsko poslovanje), nadzora dostopa (zaklenjanje avtomobilov, smučarske karte) in plačilnih postopkov (preplačniške telefonske kartice, spletna banka). Kriptografski sistem, ki je v ozadju naprav pri vseh omenjenih uporabah, mora zadoščati številnim varnostnim vidikom. Med njimi so zaupnost podatkov, celovitost podatkov, avtentikacija in nezatajljivost. Nekatere izmed teh aspektov lahko opišemo v kontekstu Boolovih funkcij.

Klasičen primer kriptosistema je prikazan na sliki 8.1. Tovrstni kriptografski gradnik predstavlja šifrirni algoritem simetričnih ključev, saj se pri šifriranju in dešifriranju uporablja enak ključ, ki si ga delita pošiljatelj in prejemnik. Kriptografija simetričnih ključev preučuje dve široki družini kriptografskih gradnikov, ki jih imenujemo bločne in tokovne šifre (glej sliko 8.2). Ker se tako bločne kot tokovne šifre precej boljše obnašajo od tehnik v kriptografiji javnih ključev, se le-te pogosto uporabljajo v praksi. Pri tem pa se dizajn enih in drugih precej razlikuje med sabo.

Pri bločnih šifrah je čistopis razdeljen na bloke (dolžina le teh je potenca števila dva, tipično 64, 128 ali 256 bitov), ki so zakodirani posamično. Dizajn enkripcijskega

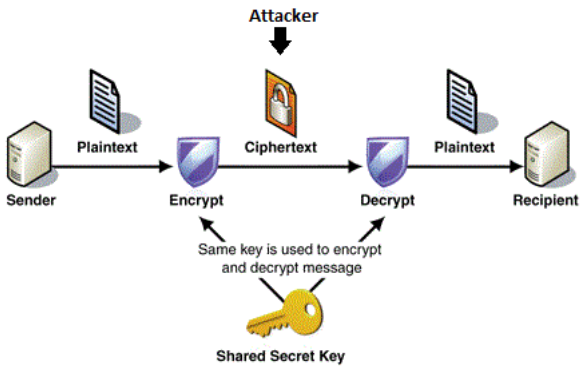

Figure 8.1: Shema klasičnega kriptosistema

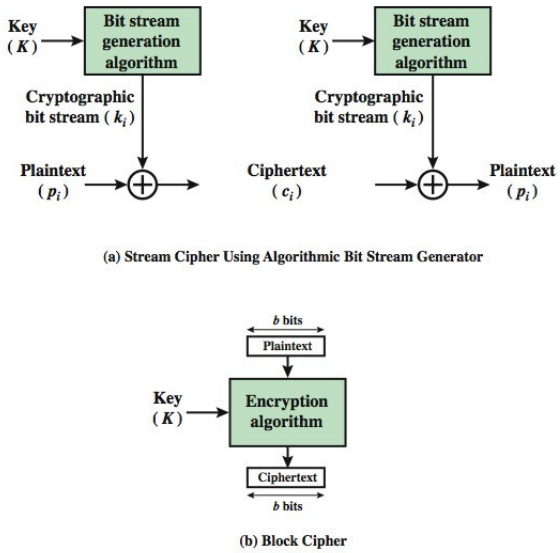

Figure 8.2: Sheme pri enkripciji simetričnih ključev

algoritma pri bločni šifri (glej sliko 8.2) uporablja kriptografske gradnike imenovane *S-škatle* (ang. substitution boxes) oz. vektorske Boolove funkcije, na katere lahko gledamo kot nabor Boolovih funkcij. Izbor in lastnosti slednjih je odvisen od posamezne uporabe. Šifriranje posameznega bloka čistopisa zajema večkratno uporabo istega sloja S-škatel v bločni šifri, kar ustreza konceptu zmede (vsak bit šifropisa je na zapleten način odvisen od čistopisa in bitov skrivnega ključa). Poleg tega se v vsaki rundi enkripcije uporabi še linearni sloj, kjer je dodan skriven ključ, kar ustreza konceptu razpršitve (biti vmesnega šifropisa so po posamezni rundi šifriranja odvisni od mnogih vhodnih podatkov). Omenjena koncepta zmede in razpršitve je vpeljal Claude E. Shannon leta 1945 v svojem delu *A Mathematical Theory of Cryptography* [105]. Čeprav ni težko zagotoviti dobrih lastnosti glede teh dveh konceptov, pa so običajno dobre tokovne šifre nekoliko hitrejše od bločnih, saj

se pri bločnih šifrah ves proces šifriranja ponovi večkrat (tipično 10 do 30 krat). Med dobro poznane bločne šifre, ki bazirajo na omrežjih Feistel ali SP (ang. Substitution Permutation), so DES (ang. Data Encryption Standard), IDEA (ang. International Data Encryption Algorithm), Triple DES, Twofish, Serpent, in AES (ang. Advanced Encryption Standard).

Na drugi strani tokovne šifre bazirajo na (ne)linearnih pomičnih registrih, ki predstavljajo naprave s končnimi stanji in so sposobne shranjevanja in manipulacije svojih bitov na (ne)linearen način. Običajno je del teh bitov sprocesiranih z (ne)linearnim mehanizmom (npr. z Boolovo filtrirno funkcijo), preko katerega nastanejo biti toka ključev. Ti so prišteti (modulo dva) k čistopisu, kar nam generira ustrezen šifropis. Za razliko od bločnih šifer je poglavitni cilj pri tokovnih šifrah bodisi hitra enkripcija (hitrejša od bločnih šifer) bodisi kompaktna strojna implementacija. Dva znana tipa pomičnih registrov sta linearni in nelinearni povratni pomični register LFSR in NFSR. Nekatere konstrukcije tokovnih šifer uporabljajo registre LFSR v kombinaciji z (vektorsko) Boolovo funkcijo, katere glavni namen je filtriranje skrivnih bitov stanj in zagotavljanje zmede pri šifri. Med pomembnejše predstavnike tokovnih šifer spadajo SEAL [95, 96], SNOW (glej npr. [36]), ISAAC [97], Grain family [47] in nekatere druge.

V splošnem nam dobre tokovne in bločne šifre zagotavljajo le računsko varnost, za razliko od kriptografskih sistemov pri kriptografiji javnih ključev, kjer je varnost povezana z določenim znanim težkim problemom, ki ga ni možno učinkovito rešiti z znanimi postopki. V nadaljevanju bomo na kratko opisali razlike med kriptografijo simetričnih ključev in kriptografijo javnih ključev. Za razliko od kriptografije simetričnih ključev, kjer isti ključ uporabljata tako pošiljatelj kot prejemnik, se pri kriptografiji javnih ključev uporablja javni ključ (poznan vsem) in zasebni ključ (poznan le prejemniku). Pri šifrirni shemi z javnim ključem lahko vsak šifrira sporočilo z javnim ključem, dešifriranje sporočila pa je možno le s prejemnikovim zasebnim ključem. Varnost pri teh sistemih se večinoma zanaša na težke matematične probleme (praštevilski razcep, problem diskretnega logaritma ipd.), za katere ni znanih učinkovitih algoritmov. Če podobne probleme, kot je problem diskretnega logaritma, definiramo na matematičnih strukturah kot so eliptične krivulje, zagotovimo še večjo varnost. Poleg tega šifrirni algoritmi iz kriptografije javnih ključev ne potrebujejo varnega kanala za začetno izmenjavo skrivnih ključev. Vendar so vsi znani kriptosistemi, ki bazirajo na javnem ključu, precej manj učinkoviti od sistemov v simetrični kriptografiji, saj je njihova podatkovna prepustnost nižja (zaradi časovne zahtevnosti enkripcije). Zaradi večje hitrosti šifriranja se sheme simetrične kriptografije uporabljajo pri šifriranju podatkov, kriptografija javnih ključev pa se uporabi za izmenjavo ključev.

Glede na tip informacije s katero razpolaga nasprotnik, ločimo štiri tipe kriptoanalize:

- Poznan šifropis - kriptoanalitik (napadalec) lahko le pasivno posluša šifrirano komunikacijo. Zgolj s pomočjo šifropisa poskuša pridobiti šifrirni ključ oz. del ključa ali del čistopisa.

- Poznan čistopis - kriptoanalitik poskuša pridobiti šifrirni ključ ali del ključa, pri čemer ima na voljo del čistopisa in pripadajoči del šifropisa.

- Izbran čistopis - kriptoanalitik lahko šifrira katerikoli izbrani čistopis. Cilj napada je pridobiti del skrivnega ključa.

- Izbran šifropis - kriptoanalitik ima napravo za dešifriranje in lahko dešifrira poljuben šifropis. Cilj napada je pridobiti ključ.

Od tu dalje se bomo osredotočili na konstrukcije in kriptoanalizo tokovnih šifer. Predvsem nas bodo zanimale tiste sheme, ki uporabljajo pomične registre LFSR/NFSR v kombinaciji s filtrirnimi (vektorskimi) Boolovimi funkcijami.

Med številnimi kriptoanalitičnimi metodami za tokovne šifre, sta *algebraični napad* (AA) in *hitri algebraični napad* (FAA) [25, 26] deležna posebne pozornosti. Ti napadi, ki so generičnega tipa za tokovne šifre, ki bazirajo na pomičnih registrih LFSR, so povečali konstrukcijske zahteve pri izbiri filtrirne (vektorske) Boolove funkcije. Osrednjo idejo pri teh dveh napadih lahko opišemo na naslednji način. V prvem koraku nastavimo sistem enačb nizke stopnje, kjer so neznanke biti skrivnega ključa, in kjer je stopnja enačb v tesni povezanosti z algebraičnimi lastnostmi filtrirne funkcije F (glej sliko 8.3). V drugem koraku rešimo sistem enačb in pridobimo bite skrivnega ključa. Medtem, ko je drugi korak dobro preučen, pa prvi korak zaradi visoke kompleksnosti predstavlja odprt problem, če je število spremenljivk n relativno veliko. V preteklem desetletju so se kriptografi in kriptoanalitiki precej ubadali z oceno zaščite nelinearne Boolove funkcije proti napadom tipov AA in FAA. Na konferenci EUROCRYPT 2003 je bil predstavljen prvi algoritem za določitev obstoja anihilatorjev stopnje d za poljubno Boolovo funkcijo f v n spremenljivkah (tj. za določitev funkcije g, za katero je $fg = 0$) [25]. Časovna zahtevnost algoritma znaša približno $O(D^3)$ operacij, kjer je $D = \sum_{i=0}^{d} \binom{n}{i}$. Sledilo je več poskusov za izboljšanje računske učinkovitosti teh ocen [3, 8, 29, 30, 56], a noben izmed predlaganih algoritmov ni dopuščal Boolovih funkcij z relativno velik številom spremenljivk, npr. $n \geq 30$. En izmed prispevkov te disertacije je učinkovita verjetnostna metoda za določitev algebraičnih lastnosti Boolovih funkcij za velik n.

Nelinearni filtrirni generator je tipični gradnik pri konstrukcijah tokovnih šifer, ki se uporabljajo pri strojni opremi (glej sliko 8.3). Sestavljen je iz enega samega

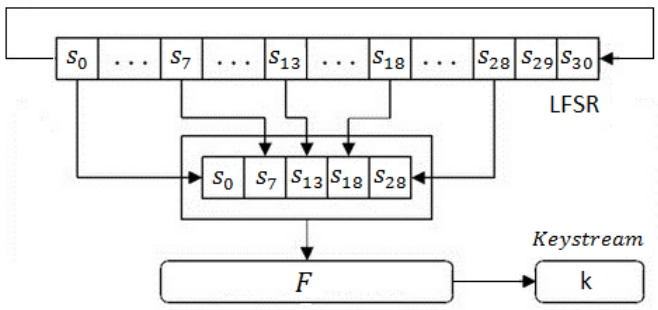

Figure 8.3: Filtrirni generator

pomičnega registra LFSR in nelinearne funkcije $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$, ki sprocesira n fiksih celic v registru LFSR. Zaščita nelinearnih filtrirnih generatorjev proti napadom kot

so (hitri) korelacijski napadi [77, 106, 82], algebraični napadi [23, 24, 78], verjetnostno algebraični napadi [9, 87] in napadi, ki izkoristijo normalnost Boolovih funkcij [83], je odvisna predvsem od izbire filtrirne funkcije F. Konstrukcijska pravila za zagotavljanje dobre zaščite proti tem napadom so tako bolj ali manj znana. Po drugi strani pa to ne velja za *kriptoanalizo tipa ugani-in-določi*, ki se večinoma ne naslanja na filtrirno funkcijo (enako velja za t.i. napade 'time-memory-data trade-off attacks' [7], [48], [54]), temveč na lastnosti registra LFSR kot so velikost, izbira primitivnega polinoma in izbira fiksnih celic. Tovrstna kriptoanaliza stremi k določanju dela skrivnih bitov v registru LFSR, pri tem pa si pomaga s strukturo šifre. Kot 'strukturo' tukaj mislimo predvsem na pozicije fiksnih celic. Pomembnost izbire slednjih je bila prvič eksplicitno zapisana v članku [42], kjer so bili vpeljani inverzni napadi (glej tudi [41, 43]). Sicer pa je ta pomembna tema o (sub)optimalni izbiri fiksnih celic, pri fiksnem številu n in dolžini L registra LFSR, precej zapostavljena v literaturi. Čeprav obstaja nekaj hevrističnih poskusov za določanje fiksnih celic, pa tovrstna učinkovita in generična metoda še ni poznana. V disertaciji bomo tako predstavili nekaj novih algoritmov za (sub)optimalno izbiro teh fiksnih celic.

Med znane metode v kriptoanalizi bločnih šifer spada tudi *diferenčna kriptoanaliza*, ki sta jo vpeljala Eli Biham in Adi Shamir [5]. Slednja se uporablja predvsem pri iteriranih bločnih šifrah, čeprav jo je moč uporabiti tudi pri določenih tokovnih šifrah. Gre za napad tipa 'izbran-čistopis', čeprav se ga lahko spremeni v napad tipa 'poznan-čistopis', v kolikor je poznanih dovolj čistopisov. V grobem rečeno ta metoda išče pare čistopisov in šifropisov, ki imajo konstantno razliko, in preučuje diferenčno obnašanje kriptosistema. V zadnjih letih je bila diferenčna kriptoanaliza posplošena na več načinov, kar vključuje prisekano analizo in diferenčno analizo višjega reda [58, 63], nemogočo diferenčno analizo [59], bumerangov napad [118] in druge.

Za zagotovitev visoke varnosti morajo funkcije, ki se uporabljajo v bločnih šifrah, zadoščati številnim kriterijem. Med mnoge pomembne kriptografske lastnosti, ki so opisane v nadaljevanju, spada tudi koncept linearne strukture. Pri funkcijah nad končnimi obsegi karakteristike dva je pomembno, da S-škatle, ki so predstavljene v obliki polinoma $F(x) \in \mathbb{F}_{2^n}[x]$, $F(x) = \sum_{i=0}^{q-1} b_i x^i$, ne premorejo elementa a za katerega velja $F(x+a)+F(x) = b$ za nek fiksen $b \in \mathbb{F}_{2^n}$ in vse $x \in \mathbb{F}_{2^n}$. Tovrstni element a se imenuje b-linearna struktura. Zato je identifikacija funkcij, ki (ne) premorejo te strukture, pomemben problem. Prve raziskave v tej smeri je objavil Evertse [37], ki je preučeval kriptoanalizo šifer tipa DES. Študij linearnih struktur zasledimo tudi pri Nybergu in Knudsenu, ki sta raziskovala varnost proti diferenčnim napadom [85] ter pri drugih kasnejših delih [64, 34, 65, 114]. Povezava med obstojem linearnih struktur in diferenčnim profilom funkcij nad končnim obsegom je pomembna pri dizajnu S-škatel. Zaradi uporabe S-škatel pri šifrah lahke uteži [55, 6] je razvoj omenjene povezave še pridobil na pomenu. Pomembnosti tovrstnih študij se je v predgovoru knjige o zlomljenih funkcijah avtorja Tokareva [117] dotaknil tudi Bart Preneel, ki je zapisal, da je morebiti največji vpliv moderne kriptografije ravno v študiju posplošitev vektorskih Boolovih funkcij, ki nudijo močno zaščito proti diferenčnim in linearnim napadom. Omenjene raziskave so namreč močno vplivale na S-škatle, ki se uporabljajo v standardu AES, kar danes uporablja milijarda naprav. Zlomljene funkcije, h katerimi se bomo posvetili kasneje, so Boolove funkcije, ki

nimajo linearnih struktur in njihovo uporabo zasledimo tako v kriptografiji, pri konstrukciji CAST, Grain in HAVAL, kot tudi v drugih področjih matematike, med katerimi so konstrukcija Hadamardovih matrik, krepko regularnih grafov, Kerdockovih kod in zaporedij CDMA.

Poleg linearnih struktur, ki smo jih omenili v kontekstu S-škatel (vektorskih Boolovih funkcij), obstajajo številni indikatorji, ki opišejo kriptografske lastnosti posamezne Boolove funkcije. Tovrstna funkcija v n spremenljivkah je preslikava iz vektorskega prostora $\mathbb{F}_2^n$ v binarni obseg $\mathbb{F}_2 = \{0, 1\}$.

Eno izmed ključnih raziskovalnih področij v kriptografiji predstavlja konstrukcija kriptografsko pomembnih Boolovih funkcij. To so funkcije, ki premorejo naslednje lastnosti. Visoka *nelinearnost* je izjemnega pomena pri dizajnu kriptosistemov simetričnih ključev, saj direktno vpliva na zaščito šifre pred številnimi metodami kriptoanalize. Le-ta izmeri Hammingovo razdaljo do množice vseh afinih funkcij. Zato visoka nelinearnost nudi večjo zaščito proti napadom afine aproksimacije [74, 75]. *Uravnoteženost* Boolove funkcije pomeni, da sta prasliki ničle in enice enako močni, kar nam zagotavlja statistično neodvisnost vhodnih in izhodnih podatkov. Visoka *algebraična stopnja* zviša linearno kompleksnost šifer. *Algebraična imunost* reda d (tj. minimalna stopnja anihilatorja dane funkcije) ima pomembno vlogo pri zaščiti proti (hitrim) algebraičnim napadom v tokovnih šifrah. Zaščita (bločne) šifre proti diferenčnim napadom je opisana z odvodi S-škatel. Visoka zaščita je zagotovljena z dobrimi diferenčnimi lastnostmi.

Največji problem pri konstrukciji kriptografsko pomembnih funkcij je v tem, da morajo biti številni, zgoraj omenjeni, kriteriji zadoščeni sočasno, pri tem pa je potrebno omeniti, da optimizacija ene lastnosti običajno pomeni zmanjšanje druge. Ker je število vseh Boolovih funkcij v n spremenljivkah izjemno veliko ($= 2^{2^n}$), celoten pregled funkcij, ki premorejo določene lastnosti, ni možen. Posledično so nove konstrukcije tovrstnih funkcij še vedno zelo zanimiv prispevek v tej raziskovalni sferi.

Pojem *zlomljene funkcije* je vpeljal Rothaus leta 1976 [98]. Gre za funkcije z maksimalno nelinearnostjo. Njihov razvoj v nadaljnjih desetletjih so podpirale številne aplikacije v različnih področjih matematike in računalništva (npr. v komunikacijskih sistemih, dizajnu zaporedij, kriptografiji, algebraičnih kodah, teoriji diferenčnih množic itd.). Obstajajo številne ekvivalentne definicije zlomljenih funkcij. Med najbolj običajne spada zgoraj omenjena, ki se nanaša na nelinearnost oz. na Hammingovo razdaljo do afinih funkcij. Slednja je v resnici povezana z ravnim Walshovim spektrom (Sylvester-Hadamardova transformacija) funkcije (glej (2.3)). Čeprav je nekaj razredov generičnih zlomljenih funkcij znanih [14, 31, 33, 60], se zdi njihova popolna klasifikacija nemogoča. Lastnost zlomljene vektorske Boolove funkcije (S-škatle) $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ lahko razširimo z zahtevo, da so vse (neničelne) linearne kombinacije koordinatnih funkcij tudi zlomljene. ˇe zapišemo funkcijo F v obliki $F(x) = (f_1(x), \ldots, f_m(x))$, kjer so f_i Boolove funkcije, to pomeni, da je funkcija $a_1 f_1(x) \oplus \ldots \oplus a_m f_m(x)$ zlomljena za vsak neničelen nabor binarnih koeficientov a_i. Konstrukcije tovrstnih vektorskih zlomljenih funkcij je prvi preučeval Nyberg v članku [84], kjer je bilo pokazano, da slednje lahko obstajajo le za $m \leq \frac{n}{2}$. Konstrukcije nekaterih tovrstnih funkcij so temeljile na določenih znanih razredih zlomljenih funkcij (npr. na razredu Maiorana-McFarland [31, 32] in Dillonovem razredu

[17, 31, 32, 98]). V kontekstu preslikav oblike $F : \mathbb{Z}_p^n \to \mathbb{Z}_p^n$, kjer je $p > 2$ praštevilo, namesto o vektorskih zlomljenih funkcijah govorimo o ravninski funkciji.

Posplošene Boolove funkcije oblike $\mathbb{Z}_q^n \to \mathbb{Z}_q$ so bile vpeljane v članku [62]. Še več preučevanja so bile deležne posplošitve tipa $\mathbb{Z}_2^n \to \mathbb{Z}_q$, kjer je $q \geq 2$ celo število, saj so naravno povezane s cikličnimi kodami nad kolobarji. V članku [101] je npr. K. U. Schmidt preučeval povezave med posplošenimi zlomljenimi funkcijami, kodami konstantne amplitude in $\mathbb{Z}_4$-linearnimi kodami ($q = 4$). Drugi tip funkcij bomo v disertaciji imenovali *posplošene zlomljene funkcije* ali *gbent funkcije*. Ostale posplošitve (zlomljenih) Boolovih funkcij najdemo v delih [101, 103, 66, 60, 109, 111, 110, 117]. Za preučevanje posplošenih zlomljenih funkcij obstaja več razlogov. V prvi vrsti so v tesnem sorodstvu s klasičnimi zlomljenimi funkcijami. Pogoj zlomljenosti komponentnih funkcij (glede na ustrezno dekompozicijo) posplošene zlomljene funkcije je bil tako preučevan v članku [109] za $q = 4$, v članku [113] za $q = 8$ in v članku [70] za $q = 16$. Zlomljenost komponentnih funkcij določenih vrst gbent funkcij je tema tudi del [107, 108, 111, 76]. Zahtevnejša naloga je podati direktno konstrukcijsko metodo, ki bi funkciji oblike $\mathbb{Z}_2^n \to \mathbb{Z}_q$ za ustrezen q priredila netrivialno dekompozicijo na standardne (še nepoznane) zlomljene funkcije. Drugi razlog za raziskovanje tovrstnih objektov je njihova neposredna povezanost z dizajnom dveh tipov komunikacijskih sistemom: OFDM (ang. Orthogonal Frequency-Division Multiplexing) [69, 94, 35] in MC-CDMA (ang. Multi-Carrier Code Division Multiple Access) [44, 45, 100]. OFDM je metoda za simultan prenos podatkov preko enakomernih nosilnih frekvenc. Uporabljena je bila za številne radijske sisteme kot so brezžično omrežja, digitalno audio in video oddajanje, internetna omrežja in 4G mobilno komuniciranje. Metoda MC-CDMA prevladuje med tretjo generacijo celičnih komunikacijskih sistemov. Predstavlja shemo z večkratnim dostopom, ki se uporablja v telekomunikacijskih sistemih, ki temeljijo na OFDM, in omogoča sočasno več uporabnikov. Obe modulacijski tehniki sta podvrženi relativno visoki vrednosti PMEPR. Golayeva zaporedja [40], ki imajo nizek PMEPR, so lahko hitro identificirana s pomočjo posplošenih Boolovih funkcij, ki so asociirana s temi zaporedji (glej članek [104] in reference v njem). Pri tem gbent funkcija ustreza zaporedju z najmanjšo vrednostjo PAPR. Zato so učinkovite konstrukcijske metode za gbent funkcije zelo uporabne v komunikacijskih sistemih.

Struktura disertacije je sledeča. V poglavju 2 predstavimo temelje (posplošenih) Boolovih funkcij in osnove kriptoanalize tipa ugani-in-določi.

V poglavju 3 je predstavljena popolna karakterizacija gbent funkcij oblike $f : \mathbb{Z}_2^n \to \mathbb{Z}_q$, kjer je q potenca števila 2. Podana je tudi analiza njihovih dualnih preslikav in Grayevih preslikav. Dokazani so tudi zadostni pogoji za posplošeno zlomljenost za poljubno sodo število q. Prikazana je povezava med določenim podrazredom gbent funkcij in relativnimi diferenčnimi množicami, ki jih imenujemo $\mathbb{Z}_q$-zlomljene funkcije. Pri tem pokažemo, da ustrezajo razredu vektorskih zlomljenih funkcij. Poglavje zaključimo s prvo splošno znano konstrukcijsko metodo za gbent funkcije.

Optimalna izbira fiksnih celic v šifrirnih shemah, ki bazirajo na pomičnih registrih LFSR je preučevana v poglavju 4. Pri tem sta obravnavana tako konstrukcijski kot tudi kriptoanalitični vidik. Predstavljena sta dva nova algoritma za optimalno izbiro fiksnih celic, ki nudita (sub)optimalno zaščito proti generičnim kriptoanal-

itičnim tehnikam za tovrstne sheme. Prikazana sta dva načina vzorčenja blokov toka ključev. Pokazano je, da se v mnogih primerih ta dva načina odrežeta bolje od standardnega načina GFSGA [119] (ki predstavlja posebno obliko kriptoanalize tipa ugani-in-določi). Pokažemo tudi, da je mogoče GFSGA napade izvesti na šifre, ki temeljijo na NFSR (npr. na šifre v [47]) in na filtrirne generatorje, ki proizvedejo posamezen bit vsakič, ko je šifra ustvarjena.

Učinkovita ocena zaščite Boolove funkcije z relativno velikim številom vhodnih spremenljivk n proti (hitrim) algebraičnim napadom je predstavljena v poglavju 5. Vpeljana je dekompozicija nelinearne Boolove funkcije na več linearnih (afinih) podfunkcij, kjer se uporabi disjunktne množice vhodnih spremenljivk. Sama dekompozicija temelji na delnih linearnih relacijah. Predstavljen je nov splošen verjetnostni dekompozicijski algoritem, ki učinkovito oceni zaščito Boolove funkcije proti (hitrim) algebraičnim napadom za velike vrednosti n. Pri tem je potrebno poudariti, da je računska kompleksnost do sedaj znanih metod previsoka za uporabo v praksi.

Disertacija se zaključi s poglavljem 6, kjer so predstavljeni številni novi neskončni razredi polinomov, ki so brez linearne strukture.

Zaključek

V disertaciji so predstavljeni številni pomembni prispevki pri reševanju odprtih problemov iz kriptografije iz zadnjih nekaj desetletij.

Večji del disertacije zajema karakterizacijo posplošenih zlomljenih funkcij (preslikav iz $\mathbb{Z}_2^n$ v $\mathbb{Z}_q$). Podani so potrebni in zadostni pogoji za tovrstne funkcije, v primeru da je q potenca števila 2. Kot direktna posledica je podana določitev dualne funkcije in analiza Grayevih preslikav. Vpeljan je pojem $\mathbb{Z}_q$-zlomljenosti, ki je analiziran. Prikazana je povezava med $\mathbb{Z}_q$-zlomljenostjo in teorijo diferenčnih množic, ki je podana preko $(2^n, 2^k, 2^n, 2^{n-k})$-relativnih diferenčnih množic v $\mathbb{F}_2^n \times \mathbb{Z}_{2^k}$. Za razliko od znanih konstrukcij gbent funkcij, ki delujejo za posamezno vrednost q, je v disertaciji podana prva generična konstrukcijska metoda.

Pomembnost optimizacije izbora fiksnih celic pri šifrah, ki temeljijo na pomičnih registrih LFSR, leži v dejstvu, da so kripto-sheme do sedaj izbirale fiksne celice predvsem preko diferenčnih množic, ali pa je bil izbor hevristične narave. Analiza GFSGA napadov z algoritmi, ki so predstavljeni v disertaciji, je pokazala, da totalno pozitivne diferenčne množice (ki se pogosto uporabljajo) ne ponujajo optimalne zaščite pri GFSGA napadih. Uporaba omenjenih algoritmov precej izboljša izbor, ki pa še ni optimalen, kar predstavlja odprt problem za nadaljnje raziskovanje. Hkrati je analiza pokazala, da ni dovolj, da izbor fiksnih celic temelji le na lastnostih porojenih iz diferenčnih množic, ampak so potrebne bolj napredne metode, ki upoštevajo naravo GSFGA napadov.

Čeprav je znanih več metod za oceno zaščite slučajne Boolove funkcije proti (hitrim) algebraičnim napadom [25, 30, 3, 8, 29, 56], so le-te v praksi običajno neuporabne, če je število n vhodnih spremenljivk relativno veliko (npr. za $n \geq 30$), saj je njihova računska kompleksnost prevelika. V disertaciji je podan učinkovit verjetnostni algoritem za oceno zaščite Boolove funkcije proti (hitrim) algebraičnim napadom, ki temelji na razrezu v parcialne linearne relacije, in ima časovno zahtevnost $O(n^2 2^n)$. V zameno za tip algoritma, ki je verjetnostne narave, tako dobimo precej

manjšo časovno zahtevnost.

Z uporabo dokaj elementarnih tehnik, ki povežejo koeficiente polinoma nad končnim obsegom in njegovega odvoda, so v disertaciji identificirani novi neskončni razredi polinomov, ki nimajo linearnih struktur. Povezava med obstojem linearnih struktur in diferenčnim profilom funkcij nad končnimi obsegi predstavlja pomembno raziskovalno področje pri konstrukciji S-škatel, saj je zagotovitev dobre zaščite proti diferenčni kriptoanalizi bistvena. Omeniti zadošča dejstvo, da nešteto današnjih aparatur uporablja enkripcijski standard AES, kar predstavlja bločno šifro, ki temelji na S-škatlah.

Osnovna orodja pri raziskovanju segajo od kombinatoričnih do algebraičnih metod iz kriptografije. Pomembno orodje pri študiju posplošenih zlomljenih funkcij je uporaba lastnosti ciklotomičnih obsegov in določenih metod iz linearne algebre. Bistveni del pri optimizaciji pozicij fiksnih celic je detajlna analiza napada GFSGA s konstantno hitrostjo vzorčenja, pomagali pa smo si tudi s sofisticiranim iskanjem s pomočjo računalniškega programa Mathematica. Pri identifikaciji novih neskončnih razredov polinomov, ki nimajo linearnih struktur, smo se posluževali teorije končnih obsegov. Razvoj algoritma, ki oceni zaščito Boolove funkcije proti napadom AA in FAA, temelji na novi metodi, ki razbije poljubno funkcijo na več majhnih delno linearnih podfunkcij, in pri tem uporabi disjunktne množice vhodnih spremenljivk.

Kazalo

Seznam slik **xi**

Seznam tabel **xiii**

1 Uvod **1**

2 Boolove funkcije in filtrirni generatorji **11**

2.1 Boolove funkcije 11
2.2 Posplošene Boolove funkcije 15
2.3 Sylvester-Hadamardova matrika 16
2.4 Nelinearen filtrirni generator 17
2.4.1 Pregled napadov FSGA in GFSGA 17

3 Posplošene zlomljene funkcije **21**

3.1 Motivacija in domneva o transformaciji GWHT 23
3.1.1 Nova formula transformacije GWHT 26
3.2 Zadostni pogoji za posplošeno zlomljenost (q sod) 30
3.2.1 Ekvivalentne forme pogojev ($\triangle$) in ($\square$) 33
3.2.2 Potrebni in zadostni pogoji za razred GMMF 35
3.2.3 Izpolnitev potrebnih pogojev za posplošeno zlomljenost 36
3.3 Popolna karakterizacija posplošenih zlomljenih funkcij 38
3.3.1 O Sylvester-Hadamardovi matriki 38
3.3.2 Potrebni in zadostni pogoji ($q = 2^k$) 40
3.3.3 Pogoji o posplošeni zlomljenosti v obliki afinih (semi-)zlomljenih prostorov 43
3.3.4 Ekvivalenca posplošenih zlomljenih funkcij 45
3.3.5 $\mathbb{Z}_q$-zlomljene funkcije in relativne diferenčne množice 48
3.3.6 Dualna in Grayeva preslikava posplošene zlomljene funkcije 50
3.3.7 Dual posplošene zlomljene funkcije 50
3.3.8 Grayeva preslikava posplošene zlomljene funkcije 51
3.4 Konstrukcijske metode za posplošene zlomljene funkcije 53
3.4.1 Opis problema 53
3.4.2 Konstrukcija posplošenih zlomljenih funkcij s pomočjo razreda MM 55
3.4.3 Semi-zlomljene funkcije disjunktnih spektrov v razredu MM 55
3.4.4 Notrivialna izbira komponentnih funkcij, n lih 57
3.4.5 Konstrukcija za sode n 61

3.4.6 Prikaz konstrukcijskih detajlov - primer 61
3.5 Konstrukcija posplošene zlomljene funkcije iz dveh (posplošenih) Boolovih funkcij 62
3.5.1 Posplošene zlomljene funkcije iz dveh zlomljenih Boolovih funkcij 64
3.5.2 Posplošene zlomljene funkcije iz direktne vsote posplošenih zlomljenih funkcij 66
3.5.3 Posplošene zlomljene funkcije iz zlomljene in posplošene zlomljene funkcije 67
3.5.4 Konkatenacija posplošenih zlomljenih Boolovih funkcij 67
3.5.5 Posplošene zlomljene funkcije na $\mathbb{Z}_2^n$, n lih 68
3.5.6 Posplošene zlomljene funkcije na $\mathbb{Z}_2^n$, n sod 71
3.5.7 Konstrukcijske metode za posplošene zlomljene funkcije v $\mathcal{GB}_n^{4s}$ 72

4 Optimizacija pozicij fiksnih celic 77

4.1 Kompleksnost proti številu ponovljenih enačb 79
4.2 Dva algoritma za optimalno izbiro fiksnih celic 83
4.3 GFSGA s spremenljivim korakom vzorčenja 88
4.3.1 Število ponovljenih enačb pri $GFSGA^*$ 90
4.3.2 Dva posebna načina $GFSGA^*$ 92
4.3.3 $GFSGA^*_{(1)}$ način napada 93
4.3.4 $GFSGA^*_{(2)}$ način napada 94
4.4 Primerjava med $GFSGA$, $GFSGA^*_{(1)}$ in $GFSGA^*_{(2)}$ 95
4.4.1 Pregled algoritmov za izbiro fiksnih celic 95
4.4.2 Totalno pozitivne diferenčne množice proti algoritmični izbiri .. 96
4.4.3 Še nekaj primerov in primerjav 98
4.5 Uporaba GFSGA v drugih primerih 100
4.5.1 Uporaba GFSGA na nelinearnih filtrirnih generatorjih z enim izhodom 100
4.5.2 Uporaba GFSGA na šifrah, ki temeljijo na NFSR 102
4.5.3 Grain-128 izbira fiksnih celic 105

5 Ocena algebraičnih lastnosti Boolovih funkcij za velike vrednosti n 107

5.1 Verjetnostni dekompozicijski algoritem za nelinearne Boolove funkcije 108
5.2 Ocena zaščite proti napadom AA in FAA 115
5.2.1 Zaščita proti AA 115
5.2.2 Zaščita proti FAA 117
5.2.3 Alogoritem za oceno zaščite proti napadom AA in FAA 119

6 O odvodih polinomov nad končnimi obsegi skozi integracijo 127

6.1 Linearne strukture in odvodi 128

6.1.1 Preliminarni rezultati s pomočjo integracijske formule 130

6.1.2 Linearne strukture preslikav nad končnimi obsegi in Boolove funkcije 131

6.2 Zgornje meje za stopnjo ravninskih preslikav 134

7 Zaključek 139

Literatura 141

8 Povzetek v slovenskem jeziku 151

8.1 Stvarno kazalo 160

Stvarno kazalo

$\mathbb{Z}_q$-zlomljenost, 18, 50

afin (semi-)zlomljen prostor, 45
afina ekvivalenca, 47
algebraična stopnja, 14
algebraična imunost, 111
algoritmi, 85
anihilator, 111
asociirana algebraična normalna forma, 14
kompleksnost napada, 21, 81

Boolova funkcija, 14
  posplošena dualna, 52
  afina funkcija, 14
  zlomljena, 16
  dualna, 17
  posplošena, 17, 25
  posplošena zlomljena (gbent), 17, 34, 43
  planotska funkcija, 17
  semi-zlomljena, 17
  vektorska, 14
  vektorska zlomljena, 16

karakterji grupe, 16

odvod, 15
disjunktni spektri, 16, 57
skalarni produkt, 14

totalno pozitivne diferenčne množice, 98

Galoisev obseg, 13
GFSGA načini
  $GFSGA^*_{(1)}$, 95
  $GFSGA^*_{(2)}$, 96
Grayeva preslikava, 53
napadi z ugibanjem, 19
  filter-state (FSGA), 19
  posplošeni filter-state (GFSGA), 20

Hammingova
  razdalja, 14
  utež, 14

leksikografsko urejanjanje, 13
linearna struktura, 131

Maiorana-McFarlandov razred, 57

nelinearen filtrirni generator, 19
nonlinearnost, 14

delne linearne relacije, 114, 117
ravninske preslikave, 138
primitivni element, 13

relativna diferenčna množica, 16, 50

zaporedje funkcije, 18
tokovne šifre
  Grain-128, 105
  SFINX, 89
  SOBER-t32, 88
nosilec funkcije, 14
Sylvester-Hadamardova matrika, 18, 40

pozicije fiksnih celic, 20, 79
sled, 15
pravilnostna tabela, 14

univariatna reprezentacija, 15

vektorski prostor, 13

Walsh-Hadamardova transformacija, 15
  posplošena, 17, 31
  normalizirana, 18

Declaration

I declare that this thesis does not contain any materials previously published or written by another person except where due reference is made in the text.

Samir Hodžić