Computational investigation of 0-APN monomials

Kjetil A. Nesheim
Supervisors: Lilya Budaghyan and Nikolay S. Kaleyski

Department of Informatics
University of Bergen

2022

Acknowledgements

I would like to extend my deepest gratitude to my supervisors Nikolay Kaleyski and Lilya Budaghyan for introducing me to the wonderful world of cryptographic Boolean functions. In particular, I would like to thank Nikolay for being so patient, providing me with ideas, sharing his enthusiasm about the subject and spending many late evenings with me in the office; just to name a few. This thesis would not be what it is today without his help. I would also like to thank the entire Boolean functions team for accompanying me to many group activities, conferences, and in general making my stay at the Selmer center a wonderful experience. Tor Helleseth also deserves a huge thanks for giving me advice and helping me formulate some of the proofs. Finally, I want to thank my parents for all the unending support, and I also want to thank my roomates Håkon, Ola and Austin for making my time in Bergen such an amazing stay the last couple of years.

Kjetil A. Nesheim

Bergen, 2022

Abstract

This thesis is dedicated to exploring methods for deciding whether a power function $F(x) = x^d$ is 0-APN. Any APN function is 0-APN, and so 0-APN-ness is a necessary condition for APN-ness. APN functions are cryptographically optimal, and are thus an object of significant interest. Deciding whether a given power function is 0-APN, or APN, is a very difficult computational problem in dimensions greater than e.g. 30. Methods which allow this to be resolved more efficiently are thus instrumental to resolving open problems such as Dobbertin's conjecture. Dobbertin's conjecture states that any APN power function must be equivalent to a representative from one of the six known infinite families. This has been verified for all dimensions up to 34, and up to 42 for even dimensions. There have, however, been no further developments, and so Dobbertin's conjecture remains one of the oldest and most well-known open problems in the area. In this work, we investigate some methods for efficiently testing 0-APN-ness.

A 0-APN function can be characterized as one that does not vanish on any 2-dimensional linear subspace. We determine the minimum number of linear subspaces that have to be considered in order to check whether a power function is 0-APN. We characterize the elements of this minimal set of linear subspaces, and formulate and implement efficient procedures for generating it. We computationally test the efficiency of this method for dimension 35, and conclude that it can be used to decide 0-APN-ness much faster than by conventional methods, although a dedicated effort would be needed to exploit this further due to the huge number of exponents that need to be checked in high dimensions such as 35. Based on our computational results, we observe that most of the cubic power functions are 0-APN. We generalize this observation into a "doubly infinite" family of 0-APN functions, i.e. a construction giving infinitely many exponents, each of which is 0-APN over infinitely many dimensions. We also present some computational results on the differential uniformity of these exponents, and observe that the Gold and Inverse power functions can be expressed using the doubly infinite family.

Introduction

In a world which has become increasingly more reliant on the Internet, the need to provide secure communications is a natural goal to have in mind. Many protocols used to provide encrypted communications over a computer network play a crucial role in our everyday lives, such as TLS and SSL, in which symmetric block ciphers play a vital part. Many block ciphers such as the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) have been developed since the 70s. Practically all such ciphers are built around mathematical objects called vectorial Boolean functions, and the properties of these functions determine the security of the cipher against attacks such as differential cryptanalysis and linear cryptanalysis. The study of Almost Perfect Nonlinear (APN) functions has in large part been motivated by their use in the design of symmetric block ciphers, as they provide optimal resistance to differential cryptanalysis.

Since they are cryptographically strong, APN functions behave unpredictably and are hard to construct and analyze. For this reason, weaker notions such as 0-APN-ness have been defined. Any APN function is 0-APN, but not necessarily vice-versa, and so 0-APN-ness constitutes a necessary condition that can sometimes be used to simplify the search for and study of APN functions.

In this thesis we will study the properties of one of the simplest classes of 0-APN functions, namely the 0-APN monomials, or power functions. A major factor motivating this study is one of the oldest open problems in the field of APN functions, namely *Dobbertin's conjecture*. Several infinite families of power APN functions have been found at the time of writing of this thesis (see Table 1.1). The conjecture states that these are the only APN monomials that exist up to equivalence. This conjecture has been verified computationally up to dimension $n = 34$, and up to $n = 42$ when n is even [5]. Proving or disproving Dobbertin's conjecture has shown itself to be a very hard mathematical problem, and the computational time when trying to find a counter-example grows exponentially with the dimension. Simply checking whether a given power func-

tion is APN for dimensions above 34 is a very resource-heavy problem, and a natural idea of making it more tractable would be to first check whether this power function is 0-APN. However, even this necessary condition for APN-ness turns out to be difficult to decide computationally. In this thesis we develop some new ideas that can be used to reduce the time needed for deciding 0-APN-ness of monomials.

In Chapter 1 we will introduce the relevant theory needed when studying APN functions, with such properties as the differential uniformity and nonlinearity. We will also look at the various ways to represent vectorial Boolean functions, such as the algebraic normal form and univariate representation, and the various equivalence relations such as CCZ-equivalence and cyclotomic equivalence that are used to classify such functions.

In Chapter 2 we investigate methods for efficiently determining whether a monomial is 0-APN. We recall that a 0-APN monomial F can be characterized as one that does not vanish on any two-dimensional linear subspace, and so determining whether F is 0-APN amounts to checking all $2^{n-1} - 1$ linear subspaces of $\mathbb{F}_{2^n}$. We observe that only a small proportion of linear subspaces need to be considered in practice when checking whether a monomial is 0-APN, and we develop theoretical results for classifying and generating this minimal set of linear subspaces.

In Chapter 3 we present a proof of concept implementation to see how well our method works in practice for dimension $n = 35$. Using the work developed in the previous chapter we implement an efficient algorithm for checking whether a function is 0-APN. While this method is much faster than checking the definition, classifying all 0-APN power functions in dimension 35 would require a dedicated effort because of the sheer amount of exponents. For this reason we restrict ourselves to power functions of algebraic degree 3, 4, 5, and $n - 3$ and $n - 4$. We determine precisely which of them are 0-APN. From these results, we observe that all of the cubics are 0-APN in dimension 35. Based on this observation, in Chapter 4 we construct an infinite family of exponents each of which is 0-APN for infinitely many dimensions n, and we compute the differential uniformity for some of these exponents in dimensions up to 13.

Contents

Chapter 1

Background

In this chapter, we give the relevant background in the study of cryptographic Boolean functions. We will introduce the concept of vectorial Boolean functions and some of their cryptographic properties, namely the differential uniformity and nonlinearity. We will then present the thesis's primary object of study: the almost perfect nonlinear (APN) functions, the partial 0-APN functions and the infinite families of power APN functions. We will then look at a long standing conjecture by Hans Dobbertin on APN power functions, which is the main motivation for this work.

1.1 Vectorial Boolean functions

An (n, m)**-function**, or equivalently a **vectorial Boolean function**, is a transformation which takes an n-dimensional binary vector as input and outputs an m-dimensional binary vector. Stated mathematically we say that a VBF (vectorial Boolean function) F is a mapping between the vector spaces $\mathbb{F}_2^n$ and $\mathbb{F}_2^m$ where $\mathbb{F}_2$ is the finite field of two elements. An $(n, 1)$-function is simply called a **Boolean function**, and an (n, m)-function F can be thought of as a vector of m Boolean functions, each of which gives one coordinate of the output. This can be done by expressing F as

$$F(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)),$$

where the Boolean functions $f_1, \ldots, f_m$ are called the **coordinate functions** of F. We call the non-zero linear combinations of a VBF's coordinate functions the **component functions** of F. We denote the component functions of an (n, m)-function F as F_b for $b \in \mathbb{F}_2^m$ and $b \neq 0$. As an example, the component functions of an $(n, 2)$-function

$F = (f_1, f_2)$, would be $F_{(1,0)} = f_1, F_{(0,1)} = f_2$ and $F_{(1,1)} = f_1 + f_2$.

One of the simplest ways to represent a VBF is simply to identify every possible input vector with its corresponding output vector in a so-called truth table.

Example 1. In the following truth table we specify a $(2, 3)$-function $F(x_1, x_2)$.

(x_1, x_2)	$F(x_1, x_2)$
$(0, 0)$	$(0, 0, 0)$
$(0, 1)$	$(0, 0, 0)$
$(1, 0)$	$(0, 0, 1)$
$(1, 1)$	$(1, 1, 0)$

Granted that n is relatively small, the simplicity of the truth table representation makes it popular when working with VBF's in low-level programming languages. To appreciate why the truth table representation does not scale well to higher dimensions, consider a $(43, 1)$-function. Its truth table would consist of 2^{43} entries of at least one bit, and at best, it would require over a terabyte of memory just to store. For this reason, other representations are used, such as for instance the algebraic normal form.

The **algebraic normal form (ANF)** of an (n, m)-function F is the multivariate polynomial

$$F(x_1, \ldots, x_n) = \sum_{I \subseteq \{1,\ldots,n\}} a_I \prod_{i \in I} x_i,$$

where a_I are vectors in $\mathbb{F}_2^m$. The degree of the ANF as a multivariate polynomial is called the **algebraic degree** of F and we denote it by $\deg(F)$.

Example 2. The $(2, 3)$-function from Example 1 can be expressed in algebraic normal form as

$$F(x_1, x_2) = (1, 1, 1)x_1x_2 + (0, 0, 1)x_1,$$

and we see that $\deg(F) = 2$, because the highest degree term in its ANF is x_1x_2.

If a function F has an algebraic degree of one or less, we say that F is **affine**. If $\deg(F) = 2$, we say the function is **quadratic**, etc. It can be shown that an affine (n, m)-function F satisfies

$$F(x + y + z) = F(x) + F(y) + F(z),$$

for any $x, y, z \in \mathbb{F}_2^n$. If we also have that $F(0) = 0$, so that $F(x+y) = F(x)+F(y)$, then we say that F is **linear**. This aligns with our intuition of linearity from other fields of mathematics.

Lastly, the representation we will be using in the remainder of this text is the **univariate** representation of an (n, n)-function F. If we identify the vector space $\mathbb{F}_2^n$ with the finite field $\mathbb{F}_{2^n}$, then we can uniquely represent the function F by the univariate polynomial

$$f(x) = \sum_{i=0}^{2^n-1} a_i x^i$$

with a_i in $\mathbb{F}_{2^n}$. The algebraic degree of a function written using the univariate representation is the maximum binary weight of an exponent of any term with a non-zero coefficient. For example, if $f(x) = x^7 + x^5 + x$, then $\deg(f) = 3$, because $7 = 2^2 + 2^1 + 2^0$. The advantage of the univariate representation is that it lends itself well to mathematical analysis, and many important vectorial Boolean functions have rather simple representations written in univariate form, while their representation in other forms is much more complicated.

Example 3. One such example with a simple univariate form is the Gold function x^3, which we will see possesses many desirable cryptographic properties over any finite field $\mathbb{F}_{2^n}$. Despite being optimal in many senses, its univariate representation consists only of a single term.

An important class of vectorial Boolean functions are the so-called **power functions (monomials)**. If for some $0 \leq d \leq 2^n - 1$ we can represent the (n, n)-function F by the univariate monomial $F(x) = x^d$, then we say that F is a power function. Monomials are particularly nice to work with, and besides the quadratic polynomials, most of the known APN functions found in the literature are monomials. Perhaps not surprisingly, due to their simple univariate representation, the earliest known examples of APN functions were actually monomials. As we will see in Section 1.4, several infinite families of such APN power functions have already been discovered.

1.2 Cryptographic properties

One of the reasons the study of vectorial Boolean functions is of interest to mathematicians and computer scientists alike, is because of their importance to cryptography.

Vectorial Boolean functions (or S-boxes) form one of the fundamental building blocks of most modern block ciphers, and if we are to ensure that the cipher is resistant to such cryptanalytic attacks as the differential and linear attack, it is important that the functions used in their design have desirable cryptographic properties. In the following section we introduce some of these important cryptographic parameters.

1.2.1 Differential Uniformity

In [1] Biham and Shamir proposed the differential attack on DES-like cryptosystems employing S-boxes in their design. In this section we will introduce the notion of the **differential uniformity** proposed by Nyberg [20], which can be seen as a measure of a function's resistance to the differential attack. For an $(n, n)-$function F, we define its derivative D_aF in direction $a \in \mathbb{F}_{2^n}$ as the function

$$D_aF(x) = F(x) + F(a + x).$$

If we let $\Delta_F(a, b)$ denote the set of solutions to the equation $D_aF(x) = b$, i.e.

$$\Delta_F(a, b) = \{x \in \mathbb{F}_{2^n} : F(x) + F(x + a) = b\},$$

then we say that the **differential uniformity** Δ_F of F is

$$\Delta_F = \max\{|\Delta_F(a, b)| : a, b \in \mathbb{F}_{2^n}, a \neq 0\}.$$

If a function F has a differential unifomity of δ, we say that F is **differentially δ-uniform**. Since we are working over a field of characteristic 2, then clearly $\Delta_F \geq 2$ for any F. A function which attains this lower bound with equality is called an **almost perfect nonlinear (APN)** function.

Remark. When we are working with monomials $F(x) = x^d$, then we can compute the differential uniformity Δ_F by only considering $a = 1$. Indeed, if we look at the equation $D_aF(x) = b$ we see that

$$x^d + (a + x)^d = b, \text{ and}$$

dividing this by a^d, we get

$$\left(\frac{x}{a}\right)^d + \left(1 + \frac{x}{a}\right)^d = \frac{b}{a^d}.$$

This is the same as saying

$$D_1F(z) = c,$$

where $z = x/a^d$ and $c = b/a^d$. Thus, if the equation $D_aF(x) = b$ has more than two solutions for some a and b, then so does the equation $D_1F(z) = c$ for some c. So, in this sense, computing the differential uniformity of a monomial is a lot easier than in the general case.

Finding and classifying APN functions is of great interest because they provide optimal resistance to differential cryptanalysis, and correspond to optimal objects in coding theory, combinatorics, sequence design, etc. As we have seen, a function F is APN if and only if the equation $D_aF(x) = b$ has at most two solutions for any $a, b \in \mathbb{F}_{2^n}$ with $a \neq 0$. An equivalent definition can be stated as follows.

(Rodier's Condition). A vectorial Boolean function F is APN if and only if all the values $x, y, z \in \mathbb{F}_{2^n}$ satisfying

$$F(x) + F(y) + F(z) + F(x + y + z) = 0,$$

belong to the curve $(x + y)(x + z)(y + z) = 0$.

Similarly to how the computation of the differential uniformity can be simplified for monomials, we can fix $x = 1$ by dividing both sides of the Rodier condition

$$x^d + y^d + z^d + (x + y + z)^d = 0$$

by x^d.

As we will see later, this alternate definition can sometimes be more useful when we are trying to study the deeper structure of APN functions. Finding APN functions is both a mathematically and computationally hard problem, which has led some authors to propose slightly more general notions that relax some of these conditions. In one such example the authors of [3] propose the notion of a (partial) x_0-APN function.

Definition 1.2.1. For some fixed $x_0 \in \mathbb{F}_{2^n}$, we say that an (n, n)-function F is **(partial)** x_0**-APN** if the points $x, y \in \mathbb{F}_{2^n}$ satisfying

$$F(x_0) + F(x) + F(y) + F(x_0 + x + y) = 0,$$

belong to the curve $(x_0 + x)(x_0 + y)(x + y) = 0$.

We will make extensive use of this definition when we study the (partial) 0-APN monomials in the remainder of this thesis.

1.2.2 Nonlinearity

Another powerful attack is the linear attack proposed by Matsui [18], in which one tries to approximate a vectorial Boolean function by a linear function. This is useful because linear functions are well understood, and can be easily analyzed using the wealth of knowledge found in linear algebra. If we are to ensure a function F is resistant to linear cryptanalysis, we would like the function F to be as *different* from any linear (or affine) function as possible. One natural way to define the difference between two functions, is using their **Hamming distance**. We define the Hamming distance between two (n, n)-functions F and G as

$$d_H(F, G) = |\{x \in \mathbb{F}_{2^n} \mid F(x) \neq G(x)\}|.$$

In [19] Nyberg proposed the idea of a function's **nonlinearity** to measure its resistance to linear cryptanalysis. The nonlinearity of an $(n, 1)$-function f is defined as

$$\mathcal{NL}(f) = \min\{d_H(f, l) : l \in \mathcal{A}_n\},$$

where $\mathcal{A}_n$ denotes the set of all affine $(n, 1)$-functions. Recall that the component functions of an (n, n)-function F consist of all the non-zero linear combinations of its coordinate functions. An attacker could succeed in finding a suitable linear approximation of a function F, if even one of its component functions is sufficiently close to an affine $(n, 1)$-function. It then becomes natural to define the nonlinearity of an (n, n)-function

as the minimum nonlinearity of all of its component functions, i.e.

$$\mathcal{NL}(F) = \min\{\mathcal{NL}(F_b) \mid b \in \mathbb{F}_2^n, b \neq 0\}.$$

It can be shown [7] that the nonlinearity of any (n, n)-function F satisfies the inequality

$$\mathcal{NL}(F) \leq 2^{n-1} - 2^{(n-1)/2}.$$

Any function that attains this upper bound with equality is called an **almost bent (AB)** function, and it provides optimal resistance to the linear attack. It should be clear from the definition that AB functions can only exist when the dimension n is odd. The optimal upper bound on the nonlinearity of functions when n is even remains an open problem. Finally, it can be shown that every AB function is also APN [7], so AB functions provide optimal resistance to both linear and differential cryptanalysis. However it is worth noting that APN functions are not always AB, even when the dimension is odd.

1.3 Equivalence of vectorial Boolean functions

Even for small values of n, conducting an exhaustive search for new APN or AB functions becomes unfeasible. To put this into perspective, the number of distinct (n, n)-functions is $(2^n)^{2^n}$ and we can immediately appreciate how badly this scales as n grows. Reducing the number of functions that need to be considered is typically done by only considering them up to a suitable notion of equivalence. In this section we will introduce three equivalence relations on vectorial Boolean functions that preserve some of the cryptographic properties discussed in the previous section. First, we will discuss what is currently known to be the most general equivalence relation that preserves differential uniformity and non-linearity, namely Carlet-Charpin-Zinoviev (CCZ)-equivalence. We will also discuss extended affine (EA)-equivalence, and finally, the specialized cyclotomic equivalence of power functions, which is the relation we will mostly concern ourselves with for the remainder of this thesis.

1.3.1 CCZ-equivalence

As stated in the previous section, Carlet-Charpin-Zinoviev equivalence is currently the most general equivalence relation on vectorial Boolean functions that preserves differential uniformity and nonlinearity. CCZ-equivalence was introduced in [6], and is defined in terms of the graphs of two functions. The graph of an (n, n)-function F is defined as the set

$$\Gamma_F = \{(x, F(x)) : x \in \mathbb{F}_{2^n}\}.$$

The elements belonging to the graph of F are pairs of elements from $\mathbb{F}_{2^n}$ so it becomes natural to identify the graph with a set of elements belonging to the finite field $\mathbb{F}_{2^{2n}}$. We say that two (n, n)-functions F and G are **CCZ-equivalent** if there exists some affine $(2n, 2n)$-permutation A that maps the graph of F to that of G, i.e. $A(\Gamma_F) = \Gamma_G$. Unlike some of the less general equivalence relations such as EA-equivalence, CCZ-equivalence does not preserve the algebraic degree of a function. This is generally desirable, since most of the APN functions documented in the literature are either monomial or quadratic, and APN functions of low algebraic degree have been shown to be vulnerable to so-called higher-order differential attacks [9]. Through CCZ-equivalence, one can usually generate equivalent APN functions of higher algebraic degree when necessary.

Since functions are only considered up to equivalence, a natural question to ask is how one would test two functions for CCZ-equivalence. In practice this is done computationally by checking the isomorphism of linear codes. Let F be an (n, n)-function. We identify the vector space $\mathbb{F}_2^n$ with the finite field $\mathbb{F}_{2^n}$ with primitive element α. We generate the $2n + 1 \times 2^n$ parity check matrix

$$P_F = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & \alpha & \dots & \alpha^{2^n-2} \\ F(0) & F(1) & F(\alpha) & \dots & F(\alpha^{2^n-2}) \end{pmatrix}$$

and its corresponding code C_F. We can then show that two (n, n)-functions F and G are CCZ-equivalent if and only if their corresponding codes C_F and C_G are permutation equivalent [13], that is, if there exists a permutation ρ of $\{1, 2, \dots, 2^n\}$, such that $(x_1, x_2, \dots, x_{2^n}) \in C_F$ if and only if $(x_{\rho(1)}, x_{\rho(2)}, \dots, x_{\rho(2^n)}) \in C_G$. This is beneficial because coding theory is a much older scientific field, and algorithms for checking per-

mutation equivalence have already been developed and implemented in various software.

1.3.2 EA-equivalence

Extended affine equivalence, or EA-equivalence for short, is a special case of CCZ-equivalence [2], and it is arguably one of the most common equivalence relations used in the study of APN and AB functions. We say that two (n, n)-functions are **EA-equivalent** if there exist three affine (n, n)-functions A, A_1, A_2 where A_1 and A_2 are permutations, such that

$$A_1 \circ F \circ A_2 + A = G. \tag{1.1}$$

It can be shown that EA-equivalence coincides with CCZ-equivalence when dealing with quadratic APN functions [21], that is, two quadratic APN functions are CCZ-equivalent if and only if they are EA-equivalent. As noted previously, almost all of the APN functions found in the literature are either monomial, quadratic or CCZ-equivalent to a quadratic function. In practice this means that we can quickly determine whether a large portion of the functions of interest are CCZ-equivalent to each other by deciding their EA-equivalence. This can be done for instance using algorithms such as [4, 14]. Some specialized cases of EA-equivalence exist. For instance, if the functions F and G satisfy (1.1) with $A = 0$, then we say that the functions are **affine equivalent**. Furthermore if A_1 and A_2 are linear, then we say that the functions are **linearly equivalent**. We only mention these for completeness and historical reasons since in the study of power APN functions it is enough to consider the following notion of equivalence.

1.3.3 Cyclotomic equivalence

Recall that a power function is one that can be expressed as x^d for some $0 \leq d \leq 2^n - 1$. As it turns out, for this special class of vectorial Boolean functions, CCZ-equivalence reduces to a specialized notion of equivalence that is relatively simple to check computationally. Let $F(x) = x^e$ and $G(x) = x^d$ be two (n, n)-power functions. We

say that F and G are **cyclotomic equivalent** if either

$$2^k e \equiv d \pmod{2^n - 1}, \text{ or}$$
$$2^k e \equiv d^{-1} \pmod{2^n - 1}$$

is satisfied for some $0 \leq k \leq n-1$. That is, if either e is in the cyclotomic coset of d modulo 2^n-1, or e is in the cyclotomic coset of d^{-1} mod 2^n-1, if it exists. It can be shown that CCZ-equivalence of power functions implies cyclotomic equivalence [22]. Checking whether two monomials are cyclotomic equivalent is a rather simple exercise. In effect, this makes it possible to check if two monomials are CCZ-equivalent in dimensions where this would otherwise be too resource-heavy using the linear code test.

1.4 Infinite families of APN power functions

One of the goals of the Boolean functions community since the 90s has been to construct infinite families of functions that are always APN over $\mathbb{F}_{2^n}$ subject to certain conditions. This is not an easy task, and it should come as no surprise that the families that have been constructed have special properties such as being quadratic or monomial. In Figure 1.1 we summarize all of the known APN monomials over $\mathbb{F}_{2^n}$.

Family	Exponent	Conditions	Algebraic Degree	Reference
Gold	$2^i + 1$	$\gcd(i, n) = 1$	2	[20]
Kasami	$2^{2i} - 2^i + 1$	$\gcd(i, n) = 1$	$i + 1$	[15]
Welch	$2^t + 3$	$n = 2t + 1$	3	[10]
Niho	$2^t + 2^{t/2} - 1$, t even $2^t + 2^{(3t+1)/2} - 1$, t odd	$n = 2t + 1$	$(t + 2)/2$ $t + 1$	[11]
Inverse	$2^{2t} - 1$	$n = 2t + 1$	$n - 1$	[20]
Dobbertin	$2^{4i} + 2^{3i} + 2^{2i} + 2^i - 1$	$n = 5i$	$i + 3$	[12]

Table 1.1: Known infinite families of APN power functions over $\mathbb{F}_{2^n}$

The most recent addition to this table was made by Hans Dobbertin in the year 2000, and it has been conjectured that these six families make up all the APN power functions that exists up to CCZ-equivalence. We will refer to to this as the **Dobbertin**

conjecture. In other words, the conjecture states that any APN monomial must be CCZ-equivalent (and hence cyclotomic equivalent) to a representative from one of the six infinite families above. Several searches have been conducted in attempts to find APN monomials that are inequivalent to the known families with no success.

We note that there are several approaches that can make the search easier. It has been shown by Dobbertin that any APN monomial x^d must satisfy $\gcd(d, 2^n - 1) = 1$ for n odd, and $\gcd(d, 2^n - 1) = 3$ for n even. We can also discard every exponent d for which x^d is not APN in a subfield $\mathbb{F}_{2^r}$ for some r dividing n. As we have seen, checking whether two monomials are CCZ-equivalent is the same as checking whether they are cyclotomic equivalent, so we can limit our search to representatives from each cyclotomic coset modulo $2^n - 1$. As it stands today, all APN exponents have been classified up to $n \leq 34$, and up to $n \leq 42$, when n is even [5]. Results like these have contributed to the *Dobbertin conjecture* being wildly considered as a hard open problem by academics in the field, and we are not aware of any further searches having been conducted. The main goal of this thesis is to evaluate the efficiency of certain methods for computationally testing the Dobbertin conjecture over finite fields of high extension degree.

Chapter 2

Contributions

Recall that Dobbertin's conjecture states that any APN power function in any dimension is equivalent to one of the six infinite families found in Table 1.1. We also recall that no counterexamples have been found for dimensions $n \leq 34$ when n is odd, and no counterexamples have been found when $n \leq 42$ and n is even. One of the general objectives when we started working on this thesis was to explore alternative ways in which one could make progress on the Dobbertin conjecture. Let us first consider what makes this a computationally hard problem. Verifying whether a function $F(x)$ is APN is done by going through all $a, b \in \mathbb{F}_{2^n}$, $a \neq 0$, and checking whether the equation

$$F(x) + F(x + a) + b = 0, \tag{2.1}$$

has more than two solutions. If we are unable to find such a pair then we have verified that $F(x)$ is APN. Just one of these tests can take upwards of several hours when the dimension is greater than 30. Secondly, in dimension 35 there are 464,637,581 exponents up to cyclotomic equivalence that are not equivalent to any of the known families. It should then be clear that conducting a search in this way is not feasible without having access to several computer cores running in parallel over a long period of time, and approaches that can potentially cut down the complexity of such a search need to be considered.

2.1 Vanishing Flats

Since doing a full exhaustive search is out of the question we start to look at alternative ways to tackle the problem. In [16] the authors provided a fresh perspective on how to

study the APN property of functions $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$. For $n \geq 2$ the authors consider the set of 2-dimensional flats in dimension n as

$$\mathcal{B}_n = \{\{x, y, z, x + y + z\} \mid x, y, z \in \mathbb{F}_{2^n}\},$$

where x, y, z and $x + y + z$ are all distinct. Now by Rodier's Condition, it is clear that F is APN if and only if

$$F(x) + F(y) + F(z) + F(x + y + z) \neq 0,$$

for all $\{x, y, z, x + y + z\} \in \mathcal{B}_n$. The authors go on to define the set of *vanishing flats* with respect to F as

$$\mathcal{VB}_{n,F} = \{\{x, y, z, x + y + z\} \in \mathcal{B}_n \mid F(x) + F(y) + F(z) + F(x + y + z) = 0\},$$

and in particular Proposition III.1 [16] gives a lower bound on the number of vanishing flats for non-APN monomials as

$$|\mathcal{VB}_{n,F}| \geq \begin{cases} \frac{2^n+1}{3} & \text{if } n \text{ is odd;} \\ \frac{2^n-1}{3} & \text{if } n \text{ is even.} \end{cases}$$

In our work, we take a similar approach, but only consider the *linear flats* on which a function F vanishes. While this only provides a necessary condition for a function to be APN (in fact, not vanishing on any linear flat means that the function is 0-APN, as we discuss in more detail below), this allows us to obtain a lower bound which is stricly better than that found in Proposition III.1. Recalling Definition 1.2.1, we say that a function F is x_0-APN if all the values $x, y \in \mathbb{F}_{2^n}$ satisfying

$$F(x) + F(y) + F(x_0) + F(x + y + x_0) = 0,$$

belong to the curve $(x + x_0)(y + x_0)(x + y) = 0$. Theorem 4.4 [3] shows that any 1-APN power function is necessarily 0-APN, and furthermore by Proposition 4.1 of [3], such a function has to be APN. As a consequence of this, we can only have monomials that are not 0-APN; ones that are 0-APN but not APN; and ones that are APN. With this in

mind, we begin by considering how monomials behave on the set of linear flats. We need the following basic definition.

Definition 2.1.1. The set of linear flats in $\mathbb{F}_{2^n}$ is

$$\mathcal{LB}_n = \{\{0, x, y, x+y\} \mid x, y \in \mathbb{F}_{2^n}^* \text{ where } x \neq y\}.$$

It is clear that we can characterize 0-APN functions as ones that do not vanish on any linear flat in $\mathcal{LB}_n$. When working with monomial functions $F(x) = x^d$ over $\mathbb{F}_{2^n}$ it is natural to only consider a special kind of flat, $\{1, c, 1+c\}$, which we will denote by $\langle c \rangle$. We will refer to flats of this form as **special linear flats** or **SLFs** for short.A special linear flat should techically contain zero, but for characterizing whether monomials vanish on it, this does not matter.

The reason that we can restrict ourselves to SLFs is the following. Let $F(x) = x^d$ vanish on the linear flat $\{0, a, b, a+b\}$, i.e.

$$F(0) + F(a) + F(b) + F(a+b) = 0,$$

that is,

$$a^d + b^d + (a+b)^d = 0.$$

We divide the above equation by a^d and we see that

$$1 + \left(\frac{b}{a}\right)^d + \left(1 + \frac{b}{a}\right)^d = 0$$

and so,

$$1 + c^d + (1+c)^d = 0,$$

where $c = b/a$. So if F vanishes on the linear flat $\{0, a, b, a+b\}$ then it also vanishes on $\langle c \rangle$. More generally, a monomial vanishes on some linear flat only if it vanishes on some SLF.

We now observe that if a monomial vanishes on some SLF $\langle c \rangle$, then it also vanishes on the SLF defined by the inverse of c. This cuts down the number of SLFs that have to be considered approximately by half.

Lemma 1. Let d be a natural number, and let $c \in \mathbb{F}_{2^n}$. If the monomial $F(x) = x^d$

vanishes on the special linear flat $\langle c \rangle$ then it also vanishes on the special linear flat $\langle c^{-1} \rangle$.

Proof. The exponent d can be written as $d = \sum 2^{a_i}$ for some natural numbers $a_1, a_2, \ldots, a_k$. If we denote $A_i = 2^{a_i}$, then we can express the exponent d of the function $F(x)$ as $d = A_1 + A_2 + \ldots + A_k$. We also denote the set $\{0, A_1, \ldots, A_k\}$ as X. The expression $1 + x^d + (x+1)^d$ is of the form

$$1 + x^d + (x+1)^d = 1 + x^d + \sum_{I \subseteq X} \prod_{j \in I} x^j \tag{2.2}$$

or simply

$$x^d + \left(x^{A_1} + \cdots + x^{A_k}\right) + \left(x^{A_1+A_2} + \cdots + x^{A_{k-1}+A_k}\right) + \cdots + x^{A_1+\cdots+A_k},$$

where x^d and $x^{A_1+\cdots+A_k}$ cancel. This is easy to see, by considering that

$$(x+1)^d = (x+1)^{A_1}(x+1)^{A_2} \cdots (x+1)^{A_k} = (x^{A_1}+1)(x^{A_2}+1)\cdots(x^{A_k}+1).$$

If we let $A_I = \sum_{i \in I} A_i$ and $B_I = \sum_{i \notin I} A_i$, where $I \subseteq X$, and substituting $x = 1/c$ in (2.2), we get

$$1 + \left(\frac{1}{c}\right)^d + \left(1 + \frac{1}{c}\right)^d = 1 + \frac{1}{c^d} + \sum_{I \subseteq X} \frac{1}{c^{A_I}}, \tag{2.3}$$

Expressing this sum under the common denominator x^d, we obtain

$$\left(1 + \frac{1}{c}\right)^d = \sum_{I \subseteq X} \frac{1}{c^{A_I}} = \sum_{I \subseteq X} \frac{c^{B_I}}{c^{B_I} c^{A_I}} = \sum_{I \subseteq X} \frac{c^{B_I}}{c^d},$$

and so (2.3) becomes

$$1 + \left(\frac{1}{c}\right)^d + \left(1 + \frac{1}{c}\right)^d = \frac{\sum_{I \subset X} c^{B_I}}{c^d},$$

where the sum on the right-hand side goes through all non-trivial subsets of X, i.e. it includes neither the empty set, nor the full set X. We note that

$$\{B_I \mid I \subset X\} = \{A_I \mid I \subset X\},$$

and so

$$\sum_{I \subset X} c^{B_I} = \sum_{I \subset X} c^{A_I},$$

and the right-hand side of the above vanishes by assumption, so

$$1 + \left(\frac{1}{c}\right)^d + \left(1 + \frac{1}{c}\right)^d = 0.$$

□

Corollary 1. If a monomial $F(x)$ vanishes on the special linear flat $\langle c \rangle = \langle 1 + c \rangle$, then it also vanishes on $\langle c^{-1} \rangle$ and $\langle (1 + c)^{-1} \rangle$.

Lemma 2. If a monomial x^d vanishes on the SLF $\langle c \rangle$, then it also vanishes on $\langle c^2 \rangle$.

Proof. By squaring the equation

$$1 + c^d + (1 + c)^d = 0,$$

we successively obtain

$$\begin{aligned} \left(1 + c^d + (1 + c)^d\right)^2 &= 0, \\ 1 + c^{2d} + (1 + c)^{2d} &= 0, \\ 1 + (c^2)^d + (1 + c^2)^d &= 0. \end{aligned}$$

□

It is thus natural to extend the notion of a cyclotomic coset to SLFs.

Definition 2.1.2. The cyclotomic coset containing the special linear flat $\langle c \rangle \bmod 2^n - 1$ is defined as the set

$$\{\langle c^{2^k} \rangle = \{1, c^{2^k}, 1 + c^{2^k}\} \mid k = 0, \ldots, n - 1\}.$$

We saw in Corollary 1 that if a monomial F vanishes on the SLF $\langle c \rangle$, then it also vanishes on $\langle c^{-1} \rangle$, $\langle (1 + c)^{-1} \rangle$ and by the previous remark it also vanishes on their entire cyclotomic cosets. This motivates the following definition.

Definition 2.1.3. Let $c \in \mathbb{F}_{2^n}$. A **wall** is the minimal set containing $\langle c \rangle$ that is closed under the operations $\langle c \rangle \mapsto \langle c^{-1} \rangle$, $\langle c \rangle \mapsto \langle (1+c)^{-1} \rangle$ and $\langle c \rangle \mapsto \langle c^2 \rangle$.

The following figure provides a visual representation of what a wall looks like. The wall consists of three "blocks" which are the cyclotomic cosets containing $\langle c \rangle, \langle c^{-1} \rangle$ and $\langle (1+c)^{-1} \rangle$. From the figure it becomes clear why we call this structure a *wall*.

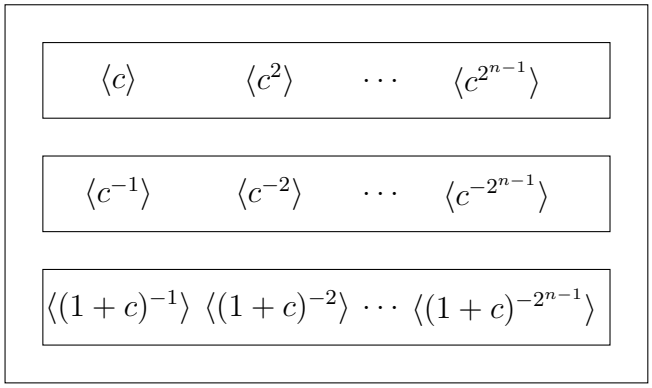

Figure 2.1: The wall containing $\langle c \rangle$

We would like to use the concept of a wall to give a lower bound on the number of SLFs on which a non-0-APN monomial vanishes. Since we know that if x^d vanishes on an SLF, then it vanishes on the entire wall containing that SLF, it would suffice for our purposes to compute the size of a wall.

Clearly, a wall can have at most $3n$ elements. However, it is possible that two of the cyclotomic cosets forming the wall will coincide with each other, or that one of the three cosets will contain less that n elements. In this case, we say that the wall "collapses"; the exact conditions under which this happens are investigated in the following section. We can thus formulate the following.

Observation 1. Let $F(x) = x^d$ be a non-0-APN monomial over $\mathbb{F}_{2^n}$. Then $F(x)$ has to vanish on at least one SLF $\langle c \rangle$. Then either $F(x)$ vanishes on at least $3n$ SLFs, or the wall containing $\langle c \rangle$ contains one or more cosets that coincide with each other, or contain fewer than n elements.

Conversely, we might wonder whether vanishing on one wall may imply vanishing on other walls. If we can prove e.g. that any non-0-APN monomial vanishes on at least 5 walls, then we can prove a much better lower bound. Unfortunately, our computational results show that the bound given by the walls is tight, i.e. there do exist monomials that vanish on one entire wall and on no other SLFs:

Observation 2. Let $F(x) = x^{1337}$ over $\mathbb{F}_{2^{17}}$ with primitive element α. The set of special linear flats on which F vanishes is of size 51. This corresponds to a single wall containing $\langle \alpha^{95} \rangle$, and it is of maximum possible size $3n$.

The notion of walls can be used to facilitate computationally checking the 0-APN property of a monomial as follows. We can partition the entire finite field into walls and we only have to test one representative from each wall. In the following table we give all the wall representatives in $\mathbb{F}_{2^n}$ where $8 \leq n \leq 11$. Here #WR denotes the number of wall representatives and #SLF denotes the number of special linear flats in dimension n.

n	Wall Representatives	#WR	#SLF
8	$\alpha, \alpha^5, \alpha^7, \alpha^9, \alpha^{13}, \alpha^{17}, \alpha^{19}, \alpha^{85}$	8	127
9	$\alpha, \alpha^3, \alpha^7, \alpha^{11}, \alpha^{17}, \alpha^{19}, \alpha^{21}, \alpha^{23}, \alpha^{27}, \alpha^{35}, \alpha^{73}$	11	255
10	$\alpha,\ \alpha^3,\ \alpha^5, \alpha^9, \alpha^{11}, \alpha^{13}, \alpha^{15}, \alpha^{21}, \alpha^{25}, \alpha^{27},$ $\alpha^{29}, \alpha^{33}, \alpha^{41}, \alpha^{47}, \alpha^{49}, \alpha^{51}, \alpha^{57}, \alpha^{73}, \alpha^{75}, \alpha^{341}$	20	511
11	$\alpha, \alpha^3, \alpha^5, \alpha^7, \alpha^{13}, \alpha^{15}, \alpha^{17}, \alpha^{19}, \alpha^{21}, \alpha^{23},$ $\alpha^{25}, \alpha^{27}, \alpha^{29}, \alpha^{33}, \alpha^{35}, \alpha^{37}, \alpha^{39}, \alpha^{43}, \alpha^{45}, \alpha^{49},$ $\alpha^{51}, \alpha^{53}, \alpha^{55}, \alpha^{67}, \alpha^{71}, \alpha^{81}, \alpha^{83}, \alpha^{99}, \alpha^{115}, \alpha^{181}, \alpha^{199}$	31	1023

Table 2.1: The wall representatives in dimensions 8 through 11.

As we can see, by pre-computing these wall representatives in $\mathbb{F}_{2^n}$ we greatly reduce the amount of elements that need to be considered when checking whether a monomial is 0-APN. For instance, in dimension 11 we will only have to check 31 elements in comparison to 1023.

2.2 The Collapsing Set

It can happen that a wall does not attain its maximum possible size of $3n$ elements; for instance, a trivial example is when the elements of $\langle c \rangle$ belong to a subfield. If we denote by A, B and C the cyclotomic coset of $\langle c \rangle, \langle c^{-1} \rangle$ and $\langle (1+c)^{-1} \rangle$ respectively, we give conditions for when the wall containing $\langle c \rangle$ might not attain its maximum possible size. These are presented in Table 2.2.

AB_0	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c = (c^{-1})^{2^k}, 0 \leq k \leq n-1\}$
AB_1	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c = 1 + (c^{-1})^{2^k}, 0 \leq k \leq n-1\}$
AC_0	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c = ((1+c)^{-1})^{2^k}, 0 \leq k \leq n-1\}$
AC_1	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c = 1 + ((1+c)^{-1})^{2^k}, 0 \leq k \leq n-1\}$
BC_0	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c^{-1} = ((1+c)^{-1})^{2^k}, 0 \leq k \leq n-1\}$
BC_1	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c^{-1} = 1 + ((1+c)^{-1})^{2^k}, 0 \leq k \leq n-1\}$
AA_1	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c = 1 + c^{2^k}, 0 \leq k \leq n-1\}$
BB_1	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c^{-1} = 1 + (c^{-1})^{2^k}, 0 \leq k \leq n-1\}$
CC_1	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)(1+c)^{-1} = 1 + ((1+c)^{-1})^{2^k}, 0 \leq k \leq n-1\}$
AA_0	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c = c^{2^k}, 0 \leq k \leq n-1\}$
BB_0	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)c^{-1} = (c^{-1})^{2^k}, 0 \leq k \leq n-1\}$
CC_0	$\{c \in \mathbb{F}_{2^n} \mid (\exists k)(1+c)^{-1} = ((1+c)^{-1})^{2^k}, 0 \leq k \leq n-1\}$

Table 2.2: The elements of the collapsing set

We will briefly discuss the naming convention of these sets. Recall that A denotes the cyclotomic coset containing $\langle c \rangle$, B denotes the cyclotomic coset containing $\langle c^{-1} \rangle$ and C denotes the cyclotomic coset containing $\langle (1+c)^{-1} \rangle$. Take for instance the set AB_1. It consists of elements $c \in \mathbb{F}_{2^n}$ such that A and B intersect with $c = 1 + (c^{-1})^{2^k}$ for some natural number k. Similarly, AB_0 consists of elements $c \in \mathbb{F}_{2^n}$ such that A and B intersect with $c = (c^{-1})^{2^k}$ for some natural number k.

Lemma 3. Some of these conditions are equivalent, namely

1. $AB_0 = CC_1$,
2. $AB_1 = AC_0 = BC_1$,
3. $AC_1 = BB_1$,
4. $BC_0 = AA_1$,
5. $AA_0 = BB_0 = CC_0$.

Proof. We start by seeing why $AB_0 = CC_1$.

1. Condition AB_0 implies

$$c + (c^{-1})^{2^k} = 0,$$

for some $0 \leq k \leq n-1$. Multiplying the above equation by c^{2^k}, we get

$$c^{2^k+1} + 1 = 0.$$

Condition CC_1 implies

$$(1+c)^{-1} + ((1+c)^{-1})^{2^k} + 1 = 0,$$

and multiplying the above by $(1+c)^{2^k+1}$, we see

$$\begin{aligned}(1+c)^{2^k} + (1+c) + (1+c)^{2^k+1} &= 0,\\ (1+c)^{2^k} + (1+c)(1+1+c)^{2^k} &= 0,\\ c^{2^k+1} + 1 &= 0,\end{aligned}$$

and it follows that $AB_0 = CC_1$.

2. Condition AB_1 implies

$$c + (c^{-1})^{2^k} + 1 = 0,$$

for some $0 \leq k \leq n-1$. Multiplying the above equation by c^{2^k} we get

$$c^{2^k+1} + c^{2^k} + 1 = 0.$$

Condition AC_0 implies,

$$c + ((1+c)^{-1})^{2^k} = 0,$$

once again multiplying by $(1+c)^{2^k}$ we see

$$c^{2^k+1} + c + 1 = 0.$$

Now if we raise this to the 2^{n-k}-th power, we see

$$\begin{aligned} c^{2^{n-k}(2^k+1)} + c^{2^{n-k}} + 1 &= 0, \\ c^{2^n+2^{n-k}} + c^{2^{n-k}} + 1 &= 0, \\ c^{2^{n-k}+1} + c^{2^{n-k}} + 1 &= 0, \\ c^{2^l+1} + c^{2^l} + 1 &= 0, \end{aligned}$$

with $l = n - k$, and it follows that $AB_1 = AC_0$. Finally, condition BC_1 implies

$$c^{-1} + ((1+c)^{-1})^{2^k} + 1 = 0,$$

multiplying this by $c(1+c)^{2^k}$ we see

$$\begin{aligned} (1+c)^{2^k} + c + c(1+c)^{2^k} &= 0, \\ 1 + c^{2^k} + c + c + c^{2^k+1} &= 0, \\ c^{2^k+1} + c^{2^k} + 1 &= 0, \end{aligned}$$

and it follows that $AB_1 = AC_0 = BC_1$.

3. Condition AC_1 implies

$$c + ((1+c)^{-1})^{2^k} + 1 = 0,$$

for some $k = 0, \ldots, n-1$. Multiplying the above equation by $(1+c)^{2^k}$ we see

$$\begin{aligned} c + c^{2^k+1} + (1+c)^{2^k} + 1 &= 0 \\ c^{2^k+1} + c^{2^k} + c &= 0. \end{aligned}$$

Condition BB_1 implies

$$c^{-1} + (c^{-1})^{2^k} + 1 = 0,$$

and multiplying this by c^{2^k+1} we get

$$c^{2^k+1} + c^{2^k} + c = 0,$$

and it follows that $AC_1 = BB_1$.

4. Condition BC_0 implies

$$c^{-1} + ((1+c)^{-1})^{2^k} = 0,$$

for some $0 \leq k \leq n-1$. Multiplying this equation by $c(1+c)^{2^k}$ we get

$$(1+c)^{2^k} + c = 0,$$
$$c^{2^k} + c + 1 = 0,$$

which obviously corresponds to AA_1, so $BC_0 = AA_1$.

□

We essentially have five conditions and their corresponding equations that can affect the size of a wall. We note that AA_0 just correspond to $\langle c \rangle$ belonging to a subfield, so we will focus on the remaining four cases:

$$AB_0 = \{c \in \mathbb{F}_{2^n} \mid (\exists k) c = (c^{-1})^{2^k}, 0 \leq k \leq n-1\} \qquad x^{2^k+1} + 1 = 0; \quad (2.4)$$
$$AB_1 = \{c \in \mathbb{F}_{2^n} \mid (\exists k) c = 1 + (c^{-1})^{2^k}, 0 \leq k \leq n-1\} \qquad x^{2^k+1} + x^{2^k} + 1 = 0; \quad (2.5)$$
$$BB_1 = \{c \in \mathbb{F}_{2^n} \mid (\exists k) c^{-1} = 1 + (c^{-1})^{2^k}, 0 \leq k \leq n-1\} \qquad x^{2^k+1} + x^{2^k} + x = 0; \quad (2.6)$$
$$AA_1 = \{c \in \mathbb{F}_{2^n} \mid (\exists k) c = 1 + c^{2^k}, 0 \leq k \leq n-1\} \qquad x^{2^k} + x + 1 = 0; \quad (2.7)$$
$$AA_0 = \{c \in \mathbb{F}_{2^n} \mid (\exists k) c = c^{2^k}, 0 \leq k \leq n-1\} \qquad x^{2^k} + x = 0. \quad (2.8)$$

This motivates the following definition.

Definition 2.2.1. The **collapsing set** $\mathcal{C}$ of $\mathbb{F}_{2^n}$ is

$$\mathcal{C} = AB_0 \cup AB_1 \cup BB_1 \cup AA_1,$$

and it consists of the elements in $\mathbb{F}_{2^n}$ representing walls of size less than $3n$.

In the following lemmas we give a series of classifications that can be used to construct the elements of the collapsing set without having to go through all elements of the finite field. This makes the collapsing set very easy to compute, even for extremely large dimensions. For the upcoming proofs we will occasionally need the following lemma from [8], and we also introduce the notation P_d to denote the set of zeroes of $x^d = 1$,

that is,

$$P_d = \{x \in \mathbb{F}_{2^n} \mid x^d = 1\}.$$

Lemma 2.1 [8]. Let $d = \gcd(\alpha, e)$, then

$$\gcd(2^\alpha + 1, 2^e - 1) = \begin{cases} 1 & \text{if } e/d \text{ is odd;} \\ 2^d + 1 & \text{if } e/d \text{ is even.} \end{cases}$$

We will now present several lemmas that can be used to classify the elements of the sets AB_0, AB_1, BB_1 and AA_1. For instance, the following lemma characterizes the case when an element from the cyclotomic coset of $\langle c \rangle$ coincides with an element from the cyclotomic coset of $\langle c^{-1} \rangle$.

Lemma 4. (AB0) Let n be even, and let $f(x) = x^{2^k+1} + 1$.

1. The set of solutions to (2.4), i.e. $f(x) = 0$ belong to the subfield $\mathbb{F}_{2^k+1}$.
2. $f(x) = 0$ has solutions if and only if $k \mid n/2$.

Proof. Assuming $k \mid n/2$, then $k \mid n$ and $\gcd(n, k) = k$. This means $n = 2bk$ for some b, and so

$$\frac{n}{\gcd(n,k)} = \frac{n}{k} = 2b,$$

and $2^k + 1 \mid 2^n - 1$ by Lemma 2.1 [8]. Conversely, assuming $2^k + 1 \mid 2^n - 1$, then $\gcd(2^k + 1, 2^n - 1) = 2^k + 1$, so $\gcd(n, k) = k$ by Lemma 2.1 and $k \mid n$. Assume to the contrary that k does not divide $n/2$, then $n = 2^e n_1$ and $k = 2^e k_1$ for some $e \geq 1$ with n_1, k_1 odd. Then $n/k = n_1/k_1$ is odd, meaning $\gcd(2^k + 1, 2^n - 1) = 1$ and consequently $2^k + 1$ does not divide $2^n - 1$ contrary to assumption. □

The following lemma handles the case when an element from the cyclotomic coset of $\langle c^{-1} \rangle$ coincides with the sum of $1 \in \mathbb{F}_{2^n}$ and another element from that same cyclotomic coset.

Lemma 5. (BB1) Let $d = \gcd(k, n)$ and let $f(x) = x^{2^k+1} + x^{2^k} + x$. Then, the set of solutions Z of equation (2.6), i.e. $f(x) = 0$ in $\mathbb{F}_{2^n}$ is

$$Z = P_{2^d+1} + 1.$$

Proof. Note that we can alternatively write the equation as

$$f(x) = (x+1)^{2^k+1} + 1 = 0.$$

The set of solutions Z of $f(x) = 0$ also satisfies $(x+1)^{2^n-1} = 1$. Therefore the set of zeros is given by

$$Z = \{x \mid (x+1)^{\gcd(2^k+1, 2^n-1)} = 1\}.$$

If $\gcd(2^k+1, 2^n-1) = 1$, then we only get the solution $x = 0$. Otherwise the set of solutions is

$$Z = \{x \mid (x+1)^{2^d+1} = 1\},$$

and $Z = P_{2^{\gcd(k,n)}+1} + 1$. □

Lemma 6. (AA1) Let $f(x) = x^{2^k} + x + 1$. The set of zeroes of (2.7), i.e. $f(x) = 0$, belong to the subfield $\mathbb{F}_{2^{\gcd(2k,n)}}$ when $\gcd(2k, n) > 1$ and $\gcd(k, n) \neq \gcd(2k, n)$.

Proof. If we raise the equation to the 2^k-th power, we see

$$x^{2^{2k}} + x^{2^k} + 1 = 0$$

but $x^{2^k} = x+1$, so $x^{2^{2k}} = x$. We note that this means that x has an order of $2^{\gcd(2k,n)} - 1$. If $\gcd(2k, n) = 1$, then x must belong to $\mathbb{F}_2$, but obviously 0 and 1 are not solutions to the equation, and furthermore if $\gcd(k, n) = \gcd(2k, n)$, then $x^{2^{2k}} + x^{2^k} = 1$ and we reach a contradiction. □

Lemma 7. (AB1) The solutions of $f(x) = x^{2^k+1} + x^{2^k} + 1 = 0$ in $\mathbb{F}_{2^n}$ belong to $\mathbb{F}_{2^{\gcd(3k,n)}}$ when $\gcd(3k, n) > 1$.

Proof. We note that $f(x) = 0$ can be written as

$$x^{2^k} = \frac{1}{x+1}.$$

If we raise this to the 2^k-th power, then we see

$$x^{2^{2k}} = \frac{1}{x^{2^k}+1} = \frac{1}{\frac{1}{x+1}+1} = \frac{x+1}{x},$$

again raising it to the 2^k-th power, we finally get

$$x^{2^{3k}} = \frac{x^{2^k}+1}{x^{2^k}} = \frac{\frac{1}{x+1}+1}{\frac{1}{x+1}} = x.$$

This means that all the zeroes of f in $\mathbb{F}_{2^n}$ are in $\mathbb{F}_{2^{\gcd(3k,n)}}$. In particular if $\gcd(3k,n)=1$, then there are no solutions in $\mathbb{F}_{2^n}$ because 0 and 1 are clearly not solutions to $f(x)=0$. □

Lemma 8. (AB1) Let $f(x)=x^{2^k+1}+x^{2^k}+1$ and let $g(x)=f(x+1)$. The equation $g(x)=0$ has 2^k+1 solutions in $\mathbb{F}_{2^{3k}}$ and the set of solutions is

$$\{x = z^{2^k-1} \mid \mathrm{Tr}_k^{3k}(z)=0, z\in\mathbb{F}_{2^{3k}}\}.$$

Proof. Substitute $x=z^{2^k-1}$ into the equation $g(x)=0$ and multiply by z. We then get that

$$zg\left(z^{2^k-1}\right) = z^{2^{2k}}+z^{2^k}+z = 0 = \mathrm{Tr}_k^{3k}(z).$$

This trace condition has 2^{2k} solutions for z in $\mathbb{F}_{2^n}$, but we note that $x=z^{2^k-1}=(az)^{2^k-1}$ for any $a\in\mathbb{F}_{2^k}$. Therefore the $2^{2k}-1$ non-zero solutions in z of $\mathrm{Tr}_k^{3k}(z)=0$, only give $(2^{2k}-1)/(2^k-1)=2^k+1$ solutions of x in $g(x)=0$. But we also note that $g(x)$ is of degree 2^k+1, so we have all the solutions to $g(x)=0$ in $\mathbb{F}_{2^{3k}}$. Finally we see that $x=z^{2^k-1}+1$ is a solution to $f(x)=0$. □

In the following table we compile some data to show how efficient the collapsing set is at eliminating non-0-APN exponents. Let SF denote the set of monomials which vanish on a subfield in dimension n, and let C denote the set of monomials that vanish on $\mathcal{C}$ in dimension n. Here the second column counts the number of distinct monomials up to cyclotomic equivalence.

n	#(Monomials)	#(Non-APN monomials)	|SF|	|C|	|C \ SF|
8	8	5	0	2	2
9	26	18	12	14	2
10	30	26	5	3	2
12	48	45	24	25	1
14	378	373	21	25	24
15	904	892	527	481	25
16	1024	1017	256	308	52
18	2592	2587	1458	1549	91
20	12000	11992	2800	1641	375
21	42340	42325	22346	21925	714
22	60016	60007	341	2636	2635

Table 2.3: Relation between the sets SF and C

When computing the collapsing set in dimension n it is beneficial to be able to compute the set of zeroes to the equation $x^d = 1$ when d divides n. This is a cyclic subgroup of $\mathbb{F}_{2^n}$, so finding a generator is enough to characterize it. Below we outline an efficient approach to finding a generator of this subgroup.

Observation 3. Let $d \mid 2^n - 1$ and let P_d denote the set of zeros to the equation $x^d = 1$. P_d is the cyclic subgroup

$$P_d = \bigcup_{m|d} \{x \mid \text{ord}(x) = m\},$$

where $\text{ord}(x)$ is the order of x.

By Theorem 1.15 (iv) [17],

$$|P_d| = \sum_{m|d} \varphi(m), \text{ where } m \mid 2^n - 1, \tag{2.9}$$

where φ is Euler's totient function, i.e.

$$\varphi(m) = |\{k \in \mathbb{N} \mid 1 \leq k < m, \gcd(k, m) = 1\}|.$$

Since P_d is cyclic we know it can be generated by α^e for some e, where α is the primitive

element in $\mathbb{F}_{2^n}$. We see that

$$|P_d| = |\{(\alpha^e)^k \mid k \in \mathbb{N}\}| = \frac{2^n - 1}{e},$$

where e is such that

$$e = \frac{2^n - 1}{|P_d|},$$

and $|P_d|$ can be computed from (2.9).

As discussed previously, Proposition III.1 [16] gives a lower bound on the number of vanishing affine flats on which a non-APN monomial vanishes. For dimension n, we have

$$|\mathcal{VB}_{n,F}| \geq \begin{cases} \frac{2^n+1}{3} & \text{if } n \text{ is odd;} \\ \frac{2^n-1}{3} & \text{if } n \text{ is even.} \end{cases}$$

The total number of affine flats in dimension n can be seen to be

$$|\mathcal{B}_n| = \frac{2^{n-2}(2^{n-1} - 1)(2^n - 1)}{3},$$

so the proportion of vanishing flats to the total number of flats is

$$|\mathcal{VB}_{n,F}|/|\mathcal{B}_n| = \frac{2^n \pm 1}{2^{n-2}(2^{n-1} - 1)(2^n - 1)},$$

and this is approximately equal to

$$\frac{1}{2^{n-2}(2^{n-1} - 1)}.$$

Using our approach, if we consider only the linear flats on which a monomial F vanishes, then we have seen by Observation 1 that either F vanishes on $\mathcal{C}$, or the number of vanishing SLFs is bounded below by $3n$. In the latter case the proportion of linear flats on which a monomial vanishes to the total number of linear flats in dimension $n > 4$ is

$$\frac{3n}{2^{n-1} - 1}.$$

This bound is better than the one in [16]. For instance, a non-0-APN monomial in dimension 9 will either vanish on $\mathcal{C}$ or it will vanish on at least 10% of all linear flats.

Chapter 3

Implementation

Now that we have introduced the relevant theoretical groundwork we can start to talk about how it can be used computationally to search for 0-APN exponents in high dimensions, and how effective this approach is in practice. We recall that the Dobbertin conjecture has been tested up to dimension 34, and so an important question is how well does this approach scale to high dimensions in practice. Most of the implementation has been done in the Magma computer algebra system, as it provides good tools for working in finite fields, but some parts have been implemented in C when we ran into time or memory constraints. All of the following computations have been run on the department server running in parallel on an Intel Xeon E5 CPU.

3.1 Overview

In the following sections, we describe the individual parts of our implementation. Our general strategy consists in performing the following steps (for some dimension n):

- We generate a list of exponents, up to cyclotomic equivalence, that are not equivalent to the known families, and may potentially be APN (that is, they do not vanish on a subfield);
- If it exists, we compute the collapsing set $\mathcal{C}$ of $\mathbb{F}_{2^n}$, and remove all exponents that vanish on $\mathcal{C}$;
- We compute the set of wall representatives of $\mathbb{F}_{2^n}$, and remove all exponents that vanish on one of them.

After this we are left with only the 0-APN monomials in $\mathbb{F}_{2^n}$.

In the following sections, we describe how each of these steps is implemented and how it performs.

3.2 Generating candidate exponents

The first thing we implemented was a way to generate all the possible candidates for new APN exponents up to cyclotomic equivalence. This was originally implemented in Magma, but in higher dimensions we had to outsource some of the computations to C for the sake of efficiency. If the dimension is even, then we only consider the exponents $2 \leq e \leq 2^n-2$, such that $\gcd(e, 2^n-1) = 3$. If the dimension is odd, then we only consider the exponents such that $\gcd(e, 2^n-1) = 1$. We then take the smallest representative from each exponent's cyclotomic coset, or possibly from the cyclotomic coset of its inverse if it has one. Then, we remove those exponents that are cyclotomic equivalent to any of the known families; this can be directly computed from the characterizations found in Table 1.1. Finally, we remove all of the exponents that vanish on a subfield of $\mathbb{F}_{2^n}$; since even for large values of n, the dimensions of the subfields are relatively small, this entire process can be performed quite fast.

3.3 Generating the set of wall representatives

Generating the wall representatives can be done fairly easily in Magma for small dimensions n. The fastest way to do this, is to begin with a set S containing all elements of $\mathbb{F}_{2^n}$. We extract an element c from S at random, generate the wall W containing c, i.e. the set consisting of the cyclotomic cosets of all of the elements

$$c, 1+c, c^{-1}, 1+c^{-1}, (1+c)^{-1}, 1+(1+c)^{-1},$$

and remove W from S. We extract the smallest element in W and add it to a set of representatives. Exactly how this minimum element is defined is not really important as long as it is consistent. In our case we used the built-in Magma function `Minimum` which orders finite field elements by their discrete logarithm.

Unfortunately, in order to take this approach, the entire field needs to be stored in memory, and this is not possible on our hardware for dimensions higher than 30.

In order to handle these cases, we take an alternative approach which is slower and somewhat more complicated, but allows us to compute the set of wall representatives without exceeding the memory limit.

The basic idea is to split the elements of $\mathbb{F}_{2^n}$ into small, manageable "chunks", and process them one at a time in Magma, by computing the wall representatives of the elements in each "chunk". These representatives are stored on the computer's hard drive, which allows the RAM to be reused for another "chunk". After all such "chunks" have been processed, the files containing the wall representatives are merged into one big file containing the representatives of the entire $\mathbb{F}_{2^n}$.

A natural question is: how to represent finite field elements when storing them in an external file. The simplest way to express an element $g \in \mathbb{F}_{2^n}$ would be to record the exponent i for which $g = \alpha^i$, where α is a primitive element of $\mathbb{F}_{2^n}$. In other words, we would record the *discrete logarithm* of g. Unfortunately, this is only possible in small dimensions; indeed, Magma only represents elements of $\mathbb{F}_{2^n}$ in terms of discrete logarithms for n up to 20. For $n > 20$, Magma instead represents elements of $\mathbb{F}_{2^n}$ using coordinate vectors, i.e. binary vectors giving the coordinates of g with respect to the standard basis of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$. We make use of this, and simply interpret the coordinate vector of g as the binary expansion of a decimal number. We then record this decimal number in the file. As an example, Magma will represent the finite field element $\alpha^{22} \in \mathbb{F}_{2^{21}}$ as $\alpha^7 + \alpha^6 + \alpha^3 + \alpha$, which we would represent by the decimal expansion

$$2^7 + 2^6 + 2^3 + 2^1 = 202.$$

We can observe that if two integers i and j belong to the same cyclotomic coset, then $\langle \alpha^i \rangle$ and $\langle \alpha^j \rangle$ belong to the same wall. Therefore, a natural first step is to partition all possible exponents i between 0 and $2^n - 2$ into cyclotomic cosets. For the sake of efficiency, we do this in C.

We then split the cosets into smaller "chunks", and process each "chunk" in Magma. More precisely, for every integer i, we generate the wall containing α^i, and we output the integer representation (as discussed above) of its smallest element into an external file.

Using this approach it is possible that we encounter the same representative in multiple files. We dealt with this problem by first using the UNIX command `sort` to sort

the files individually. We then used the command `sort -m` to merge all of our sorted "chunks" into one big file, and finally we used `uniq` to remove the duplicates that might have been introduced by using this approach.

3.4 Generating the collapsing set

- We will first tackle how to generate $AB0$ (2.4) and $BB1$ (2.6). Recall that we denote P_d as the set $P_d = \{x \in \mathbb{F}_{2^n} \mid x^d = 1\}$, and by Observation 3 we have a method to compute a generator of P_d. That is, we can compute the exponent, e, of the generator α^e of P_d by taking $e = (2^n - 1)/|P_d|$. We can efficiently compute the size of P_d in Magma by considering the identity

$$|P_d| = \sum_{m|d} \varphi(m), \text{ where } m \mid 2^n - 1.$$

  In the case of $AB0$ we can choose k such that $2k \bmod n = 0$ according to Lemma 4, and take the union of several sets P_{2^k+1}. Recall that 2.4 and 2.6 are the same with x subsituted for $x + 1$, so we can use the same approach for $BB1$.

- By Lemma 7 the set of solutions of (2.5), i.e., $AB1$, belong to the subfield $\mathbb{F}_{2^{\gcd(3k,n)}}$, and the non-trivial solutions belong to $\mathbb{F}_{2^n}$ when $n = 3k$. By Lemma 8 we know that these are $(2^k - 1)$-th powers satisfying $x = z^{2^k-1}$ with trace $\mathrm{Tr}_k^n(z) = 0$. We generate the $2^k - 1$-th powers as $x = \alpha^{(2^k-1)i}$ $(i \in \mathbb{N})$, and check whether $\mathrm{Tr}_k^n(\alpha^i) = 0$. If the trace condition is satisfied we append x and $x + 1$ to the set $AB1$.

- By Lemma 6 we know that the set $AA1$ (2.7) belongs to the subfield $\mathbb{F}_{2^{\gcd(2k,n)}}$, and the solutions of interest are those which belong to $\mathbb{F}_{2^n}$ where $n = 2k$. We can also tell from the definition that $x^{2^k} + x = 1$, meaning $\mathrm{Tr}_k^{2k}(x) = 1$. The elements x of the half-field, that is $x \in \mathbb{F}_{2^{n/2}}$, satisfy $\mathrm{Tr}(x) = 0$, so we generate these as powers of α^{2^k+1}, and having generated this we search for a single element a such that $\mathrm{Tr}(a) = 1$. Our solution set is then the additive coset

$$AA1 = \{x \in \mathbb{F}_{2^n} \mid \mathrm{Tr}_k^n(x) = 0\} + a.$$

3.5 Computational observations

Generating the wall representatives in dimension $n = 35$ took about 9 to 12 days, and we were left with an approximately 2 GB large file consisting of 166,489,130 wall representatives. We decided to split this file into sixteen different files consisting of ten million wall representatives each, so that we could run several tests in parallel. Generating the exponents up to cyclotomic equivalence takes around 10 minutes using the C program. These are then loaded into Magma where we remove all exponents that are cyclotomic equivalent to the known families of which there are 1855, and so we have a 4.5 GB file consisting of 464,635,753 exponents which need to be checked. Checking whether one of these exponents vanishes on any of the wall representatives in dimension 35 takes roughly 3 minutes running in parallel on ten million wall representatives each. It is then clear that even using the wall construction, conducting a full search for 0-APN monomials in dimension 35 is not feasible, and a dedicated effort would be needed to exploit this further.

With this in mind, we decided to limit the number of exponents we have to consider by only looking at monomials which are cyclotomic equivalent to a monomial of algebraic degree 3 or $n-3$ (which we call anti-cubics), 4 or $n-4$ (anti-quartics) and 5. The exponents were generated in the following way: we generate all possible exponents of given algebraic degree, we remove the ones that are cyclotomic equivalent to one of the known families, and finally we remove the ones that vanish on the subfields $\mathbb{F}_{2^7}$ and $\mathbb{F}_{2^5}$. These tests were run over the course of three weeks, and we noticed some interesting results. In the case of the cubics and *anti-cubics* we only have to consider 153 cubic, and 184 anti-cubic exponents up to cyclotomic equivalence. Interestingly we were not able to eliminate any cubic exponents, meaning all of the cubic monomials in dimension 35 are 0-APN. However, out of all monomials cyclotomic equivalent to ones of algebraic degree $n-3$ we were able to verify that the two exponents 1799777019 and 6239770235 are not 0-APN. For the quartics and *anti-quartics* we only need to consider 1279 quartic, and 1137 anti-quartic exponents (up to cyclotomic equivalence), and we were able to verify

that the exponents

$$548865, 136193, 12545, 280577, 1327105, 557569, 6860173,$$
$$704111761, 7879741117, 1646159603, 894429997, 7879741117,$$

are not 0-APN. In the case of quintics we only need to consider 7297 exponents up to cyclotomic equivalence, and we were able to verify that

$$2179585, 787201, 4202627, 2150401, 8913425, 8585345,$$
$$8408065, 69214337, 278577, 4718723, 1131521, 2097701,$$
$$1441801, 198145, 1073409, 2101393, 37905, 2170889, 3601,$$
$$17827905, 32977, 4198411, 819713, 11777, 2115589, 68161665,$$
$$135685, 1118211, 33687617, 17834001, 65873, 98325,$$

are not 0-APN in dimension 35.

The relevant Magma and C code can be found at https://github.com/Omeletil/OAPNTest/, together with some more details on the computational results.

Chapter 4

A doubly infinite family of 0-APN monomials

As noted in the previous chapter, one of our interesting computational results was the fact that all of the cubic monomials in dimension 35 are 0-APN. Naturally this lead us to consider what is special about these exponents. In [3] the authors prove that the function $F(x) = x^{21}$ is 0-APN if and only if n is not a multiple of 6. We note that $F(x)$ is cubic, and the exponent is equal to $21 = 2^4 + 2^2 + 1$. This appears to be a nice structure worthy of further investigation. This lead us to cubic exponents of the form $e_k = 2^{2k} + 2^k + 1$ for some natural number k, and we were able to prove the following:

Lemma 9. Let $F(x) = x^{2^{2k}+2^k+1}$. If n and k are natural numbers such that $\gcd(3k, n) = \gcd(2k, n) = 1$, then the cubic monomial $F(x)$ is 0-APN in $\mathbb{F}_{2^n}$.

Proof. We consider the equation characterizing the 0-APN-ness of $F(x)$, that is

$$F(a) + F(a + x) + F(x) = 0.$$

In the case of monomials, this becomes

$$1 + F(x) + F(x + 1) = 0.$$

Denoting the expression on the left-hand side above by $D(x)$, we see that $D(x) = 0$ can be expressed as

$$x^{2^{2k}+2^k+1} + (1 + x)^{2^{2k}+2^k+1} + 1 = 0,$$
$$x^{2^{2k}+2^k} + x^{2^{2k}+1} + x^{2^{2k}} + x^{2^{2^k}+1} + x^{2^k} + x = 0. \tag{4.1}$$

By raising the above equation to the 2^k-th power, we get that

$$x^{2^{3k}+2^{2k}} + x^{2^{3k}+2^k} + x^{2^{3k}} + x^{2^{2^{2k}+2^k}} + x^{2^{2k}} + x^{2^k} = 0. \tag{4.2}$$

Adding equations (4.1) and (4.2), we see that

$$\begin{aligned} x^{2^{3k}+2^{2k}} + x^{2^{3k}+2^k} + x^{2^{3k}} + x^{2^{2k}+1} + x^{2^k+1} + x = 0, \\ x^{2^{3k}}\left(x^{2^{2k}} + x^{2^k} + 1\right) + x\left(x^{2^{2k}} + x^{2^k} + 1\right) = 0, \end{aligned}$$

which can be factored as

$$\left(x^{2^{3k}} + x\right)\left(x^{2^{2k}} + x^{2^k} + 1\right) = 0.$$

This can have non-trivial solutions if

$$x^{2^{3k}} + x = 0,$$

meaning $x \in \mathbb{F}_{2^{\gcd(3k,n)}}$. However if we look at

$$x^{2^{2k}} + x^{2^k} + 1 = 0, \tag{4.3}$$

then we see that by raising (4.3) to the 2^k-th power, we get

$$x^{2^{3k}} + x^{2^{2k}} + 1 = 0. \tag{4.4}$$

Adding equations (4.3) and (4.4), we have

$$\begin{aligned} x^{2^{3k}} + x^{2^k} = 0, \\ x^{2^{2k}} + x = 0, \end{aligned}$$

meaning $x \in \mathbb{F}_{2^{\gcd(2k,n)}}$. Combining the above, we see that if $\gcd(3k, n) = \gcd(2k, n) = 1$, then we only have the trivial solutions $x \in \mathbb{F}_2$ and F is 0-APN. □

However, this can be generalized further. Let $e(l, k) = \sum_{i=0}^{l-1} 2^{ik}$ for some natural numbers k and l. For any choice of l and k, we give a list of dimensions n over which

$x^{e(l,k)}$ is 0-APN. In this way, we construct infinitely many monomials $F(x) = x^{e(l,k)}$, each of which is 0-APN for infinitely many dimensions n. In this sense we define a "doubly infinite" family of 0-APN monomials.

Theorem 1. Let n, l, k be natural numbers such that $\gcd(kl, n) = 1$ and

$$\gcd(e(k, l-1), 2^n - 1) = 1,$$

then $x^{e(l,k)}$ is 0-APN over $\mathbb{F}_{2^n}$.

Proof. Denote $e = e(l, k)$. Suppose that $x \in \mathbb{F}_{2^n}$ satisfies $x^e + (x+1)^e + 1 = 0$. For natural numbers $a \leq b$ and a set I, let $[a, b] = \{a, a+1, \ldots, b\}$, and let $\mathcal{P}I$ denote the power set of I. Furthermore, let $x^{2^{kI}}$ denote $\prod_{i \in I} x^{2^{ki}}$. Then $x^e + (x+1)^e + 1 = 0$ can be written as

$$x^e + \sum_{I \in \mathcal{P}[0,l-1]} x^{2^{kI}} + 1 = \sum_{\substack{I \in \mathcal{P}[0,l-1] \\ I \neq \emptyset, [0,l-1]}} x^{2^{kI}} = 0. \tag{4.5}$$

Raising this to the power 2^k yields

$$\sum_{\substack{I \in \mathcal{P}[1,l] \\ I \neq \emptyset, [1,l]}} x^{2^{kI}} = 0.$$

Adding the two expression above together causes all terms $x^{2^{kI}}$ corresponding to subsets I that contain neither 0 nor l to cancel out, leaving us with

$$\sum_{\substack{I \in (\{0\} \cup \mathcal{P}[1,l-1]) \\ I \neq [1,l-1]}} x^{2^{kI}} + \sum_{\substack{I \in (\mathcal{P}[1,l-1] \cup \{l\}) \\ I \neq [1,l-1]}} x^{2^{kI}} = 0.$$

This then becomes

$$x \left(\sum_{\substack{I \in \mathcal{P}[1,l-1] \\ I \neq [1,l-1]}} x^{2^{kI}} \right) + x^{2^{lk}} \left(\sum_{\substack{I \in \mathcal{P}[1,l-1] \\ I \neq [1,l-1]}} x^{2^{kI}} \right) = \left(x + x^{2^{lk}} \right) \left(\sum_{\substack{I \in \mathcal{P}[1,l-1] \\ I \neq [1,l-1]}} x^{2^{kI}} \right) = 0.$$

If $x + x^{2^{lk}} = 0$, then we must have $x \in \mathbb{F}_{2^{\gcd(n,lk)}}$. However, by assumption, $\gcd(n, lk) = 1$,

and so $x \in \mathbb{F}_2$. If $x \neq x^{2^{lk}}$, then we must have

$$\left(\sum_{\substack{I \in \mathcal{P}[1,l-1] \\ I \neq [1,l-1]}} x^{2^{kI}} \right) = \left(\sum_{\substack{I \in \mathcal{P}[0,l-2] \\ I \neq [0,l-2]}} x^{2^{kI}} \right)^{2^k} = 0$$

instead. Comparing this with (4.5), we see that this is simply

$$(x^{e(l-1,k)} + (x+1)^{e(l-1,k)})^{2^k} = 0,$$

and hence

$$x^{e(l-1,k)} + (x+1)^{e(l-1,k)} = 0. \tag{4.6}$$

Assuming $x \neq 0$, the above implies $(\frac{x}{x+1})^{e(l-1,k)} = 1$. If the second condition of the hypothesis is satisfied, i.e. $\gcd(e(l-1,k), 2^n - 1) = 1$, then we immediately have $\frac{x}{x+1} = 1$, i.e. $x = x + 1$, which is impossible. Therefore, $x^{e(l,k)}$ is 0-APN. □

Remark. The proof above could have also been continued by adding (4.6) to its 2^k-th power; this would have produced the same equation as if we had added the derivative $x^{e(l-1,k)} + (x+1)^{e(l-1,k)} + 1$ to its 2^k-th power since the extra term 1 cancels out. By induction on l, we would have obtained the condition that if $\gcd(ik, n) = 1$ for $i = 2, 3, \ldots, l$, then $x^{e(k,l)}$ must be 0-APN. We have tested these conditions computationally, and have seen that the condition in the statement of Theorem 1 always produces a set of dimensions n that subsumes those given by the alternative condition described in this remark. We have thus formulated the theorem only in terms of this more general condition. The less general condition $\gcd(ik, n) = 1$ for $i = 2, 3, \ldots, l$ could be useful in some contexts, however, since it does not require the explicit computation of the exponent $e(l, k)$ and its GCD with $2^n - 1$.

Remark. We can see that representatives from some of the known infinite families of APN monomials can be expressed in the form $e(l, k)$. The Gold functions x^{2^k+1} can clearly be expressed as $e(2, k)$. The inverse function can be written as $e(n-1, 1) = \sum_{i=0}^{n-2} 2^i = 2^{n-1} - 1$. We have also observed that in some cases, e.g. for $l = (n-1)/2$ and $k = 2$, or for $l = (n-1)/2 + 1$ and $k = 1$, $e(l, k)$ is equivalent to a Gold function. We leave the characterization of cases when $e(l, k)$ is equivalent to the known APN families

as a problem for future work.

In the following table we summarize the differential uniformity of $F(x) = x^{e(l,k)}$ for suitable l and k in dimensions 8 through 13, and note when F is equivalent to a monomial from a known family. Here Δ_F in the fourth column denotes the differential uniformity of F. The fifth column lists the entire cyclotomic coset of $e(l, k)$. Finally, the last column indicates which monomial family the exponent $e(l, k)$ belongs to in the case when it is APN. We note that when the dimension is odd, our construction can be equivalent to a Gold function, the inverse of a Gold function and the inverse function. In the even case, we can only obtain a Gold function in the case that $x^{e(l,k)}$ is APN. We note that the inverse function can still be expressed as $x^{e(n-1,1)}$ in the case of even dimensions, although then this function is not APN (in fact, it is differentially 4-uniform), and so we do not indicate it in the last column.

$\mathbb{F}_{2^n}$	(l,k)	$e(l,k)$	Δ_F	Cyclotomic Coset	Family
$\mathbb{F}_{2^8}$	$(2,1)$	3	2	$\{3,6,12,24,48,96,129,192\}$	Gold
	$(2,3)$	9	2	$\{9,18,33,36,66,72,132,144\}$	Gold
	$(2,2)$	5	4	$\{5,10,20,40,65,80,130,160\}$	
	$(3,2)$	21	4	$\{21,42,69,81,84,138,162,168\}$	
	$(7,1)$	127	4	$\{127,191,223,239,247,251,253,254\}$	
	$(3,1)$	7	6	$\{7,14,28,56,112,131,193,224\}$	
	$(6,1)$	63	6	$\{63,126,159,207,231,243,249,252\}$	
	$(4,1)$	15	14	$\{15,30,60,120,135,195,225,240\}$	
	$(5,1)$	31	16	$\{31,62,124,143,199,227,241,248\}$	
$\mathbb{F}_{2^9}$	$(2,1)$	3	2	$\{3,6,12,24,48,96,192,257,384\}$	Gold
	$(2,2)$	5	2	$\{5,10,20,40,80,129,160,258,320\}$	Gold
	$(2,4)$	17	2	$\{17,33,34,66,68,132,136,264,272\}$	Gold
	$(5,1)$	31	2	$\{31,62,124,248,271,391,451,481,496\}$	Gold
	$(8,1)$	255	2	$\{255,383,447,479,495,503,507,509,510\}$	Inverse
	$(3,1)$	7	6	$\{7,14,28,56,112,224,259,385,448\}$	
	$(3,2)$	21	6	$\{21,42,84,133,161,168,266,322,336\}$	
	$(6,1)$	63	6	$\{63,126,252,287,399,455,483,497,504\}$	
	$(2,3)$	9	8	$\{9,18,36,65,72,130,144,260,288\}$	
	$(4,1)$	15	8	$\{15,30,60,120,240,263,387,449,480\}$	
	$(4,2)$	85	8	$\{85,149,165,169,170,298,330,338,340\}$	
	$(7,1)$	127	8	$\{127,254,319,415,463,487,499,505,508\}$	
$\mathbb{F}_{2^{10}}$	$(2,1)$	3	2	$\{3,6,12,24,48,96,192,384,513,768\}$	Gold
	$(2,3)$	9	2	$\{9,18,36,72,129,144,258,288,516,576\}$	Gold
	$(2,2)$	5	4	$\{5,10,20,40,80,160,257,320,514,640\}$	
	$(2,4)$	17	4	$\{17,34,65,68,130,136,260,272,520,544\}$	
	$(3,2)$	21	4	$\{21,42,84,168,261,321,336,522,642,672\}$	
	$(9,1)$	511	4	$\{511,767,895,959,991,1007,1015,1019,1021,1022\}$	
	$(3,1)$	7	6	$\{7,14,28,56,112,224,448,515,769,896\}$	
	$(3,3)$	73	6	$\{73,137,145,146,274,290,292,548,580,584\}$	
	$(4,1)$	15	6	$\{15,30,60,120,240,480,519,771,897,960\}$	
	$(7,1)$	127	6	$\{127,254,508,575,799,911,967,995,1009,1016\}$	
	$(8,1)$	255	6	$\{255,510,639,831,927,975,999,1011,1017,1020\}$	
	$(4,2)$	85	10	$\{85,170,277,325,337,340,554,650,674,680\}$	
	$(5,1)$	31	30	$\{31,62,124,248,496,527,775,899,961,992\}$	
	$(6,1)$	63	32	$\{63,126,252,504,543,783,903,963,993,1008\}$	
$\mathbb{F}_{2^{11}}$	$(2,1)$	3	2	$\{3,6,12,24,48,96,192,384,768,1025,1536\}$	Gold
	$(2,2)$	5	2	$\{5,10,20,40,80,160,320,513,640,1026,1280\}$	Gold
	$(2,3)$	9	2	$\{9,18,36,72,144,257,288,514,576,1028,1152\}$	Gold
	$(2,4)$	17	2	$\{17,34,68,129,136,258,272,516,544,1032,1088\}$	Gold
	$(2,5)$	33	2	$\{33,65,66,130,132,260,264,520,528,1040,1056\}$	Gold
	$(6,1)$	63	2	$\{63,126,252,504,1008,1055,1551,1799,1923,1985,2016\}$	Gold
	$(10,1)$	1023	2	$\{1023,1535,1791,1919,1983,2015,2031,2039,2043,2045,2046\}$	Inverse
	$(3,1)$	7	6	$\{7,14,28,56,112,224,448,896,1027,1537,1792\}$	
	$(3,2)$	21	6	$\{21,42,84,168,336,517,641,672,1034,1282,1344\}$	
	$(3,3)$	73	6	$\{73,146,265,289,292,530,578,584,1060,1156,1168\}$	
	$(4,1)$	15	6	$\{15,30,60,120,240,480,960,1031,1539,1793,1920\}$	
	$(4,2)$	85	6	$\{85,170,340,533,645,673,680,1066,1290,1346,1360\}$	
	$(5,1)$	31	6	$\{31,62,124,248,496,992,1039,1543,1795,1921,1984\}$	
	$(5,2)$	341	6	$\{341,597,661,677,681,682,1194,1322,1354,1362,1364\}$	
	$(7,1)$	127	6	$\{127,254,508,1016,1087,1567,1807,1927,1987,2017,2032\}$	
	$(8,1)$	255	6	$\{255,510,1020,1151,1599,1823,1935,1991,2019,2033,2040\}$	
	$(9,1)$	511	6	$\{511,1022,1279,1663,1855,1951,1999,2023,2035,2041,2044\}$	

Table 4.1: Continued on the next page

$\mathbb{F}_{2^n}$	(l,k)	$e(l,k)$	Δ_F	Cyclotomic Coset	Family
$\mathbb{F}_{2^{12}}$	$(2,1)$	3	2	$\{3,6,12,24,48,96,192,384,768,1536,2049,3072\}$	Gold
	$(2,5)$	33	2	$\{33,66,129,132,258,264,516,528,1032,1056,2064,2112\}$	Gold
	$(2,2)$	5	4	$\{5,10,20,40,80,160,320,640,1025,1280,2050,2560\}$	
	$(3,3)$	73	4	$\{73,146,292,521,577,584,1042,1154,1168,2084,2308,2336\}$	
	$(11,1)$	2047	4	$\{2047,3071,3583,3839,3967,4031,4063,4079,4087,4091,4093,4094\}$	
	$(3,1)$	7	6	$\{7,14,28,56,112,224,448,896,1792,2051,3073,3584\}$	
	$(5,2)$	341	6	$\{341,682,1109,1301,1349,1361,1364,2218,2602,2698,2722,2728\}$	
	$(2,3)$	9	8	$\{9,18,36,72,144,288,513,576,1026,1152,2052,2304\}$	
	$(10,1)$	1023	8	$\{1023,2046,2559,3327,3711,3903,3999,4047,4071,4083,4089,4092\}$	
	$(4,2)$	85	10	$\{85,170,340,680,1045,1285,1345,1360,2090,2570,2690,2720\}$	
	$(4,1)$	15	14	$\{15,30,60,120,240,480,960,1920,2055,3075,3585,3840\}$	
	$(8,1)$	255	14	$\{255,510,1020,2040,2175,3135,3615,3855,3975,4035,4065,4080\}$	
	$(2,4)$	17	16	$\{17,34,68,136,257,272,514,544,1028,1088,2056,2176\}$	
	$(5,1)$	31	16	$\{31,62,124,248,496,992,1984,2063,3079,3587,3841,3968\}$	
	$(9,1)$	511	16	$\{511,1022,2044,2303,3199,3647,3871,3983,4039,4067,4081,4088\}$	
	$(3,2)$	21	20	$\{21,42,84,168,336,672,1029,1281,1344,2058,2562,2688\}$	
	$(6,1)$	63	62	$\{63,126,252,504,1008,2016,2079,3087,3591,3843,3969,4032\}$	
	$(7,1)$	127	64	$\{127,254,508,1016,2032,2111,3103,3599,3847,3971,4033,4064\}$	
$\mathbb{F}_{2^{13}}$	$(2,1)$	3	2	$\{3,6,12,24,48,96,192,384,768,1536,3072,4097,6144\}$	Gold
	$(2,2)$	5	2	$\{5,10,20,40,80,160,320,640,1280,2049,2560,4098,5120\}$	Gold
	$(2,3)$	9	2	$\{9,18,36,72,144,288,576,1025,1152,2050,2304,4100,4608\}$	Gold
	$(2,4)$	17	2	$\{17,34,68,136,272,513,544,1026,1088,2052,2176,4104,4352\}$	Gold
	$(2,5)$	33	2	$\{33,66,132,257,264,514,528,1028,1056,2056,2112,4112,4224\}$	Gold
	$(2,6)$	65	2	$\{65,129,130,258,260,516,520,1032,1040,2064,2080,4128,4160\}$	Gold
	$(7,1)$	127	2	$\{127,254,508,1016,2032,4064,4159,6175,7183,7687,7939,8065,8128\}$	Gold
	$(12,1)$	4095	2	$\{4095,6143,7167,7679,7935,8063,8127,8159,8175,8183,8187,8189,8190\}$	Inverse
	$(3,1)$	7	6	$\{7,14,28,56,112,224,448,896,1792,3584,4099,6145,7168\}$	
	$(3,2)$	21	6	$\{21,42,84,168,336,672,1344,2053,2561,2688,4106,5122,5376\}$	
	$(3,3)$	73	6	$\{73,146,292,584,1033,1153,1168,2066,2306,2336,4132,4612,4672\}$	
	$(3,4)$	273	6	$\{273,529,545,546,1058,1090,1092,2116,2180,2184,4232,4360,4368\}$	
	$(4,1)$	15	6	$\{15,30,60,120,240,480,960,1920,3840,4103,6147,7169,7680\}$	
	$(4,3)$	585	6	$\{585,1097,1161,1169,1170,2194,2322,2338,2340,4388,4644,4676,4680\}$	
	$(5,1)$	31	6	$\{31,62,124,248,496,992,1984,3968,4111,6151,7171,7681,7936\}$	
	$(6,1)$	63	6	$\{63,126,252,504,1008,2016,4032,4127,6159,7175,7683,7937,8064\}$	
	$(6,2)$	1365	6	$\{1365,2389,2645,2709,2725,2729,2730,4778,5290,5418,5450,5458,5460\}$	
	$(8,1)$	255	6	$\{255,510,1020,2040,4080,4223,6207,7199,7695,7943,8067,8129,8160\}$	
	$(9,1)$	511	6	$\{511,1022,2044,4088,4351,6271,7231,7711,7951,8071,8131,8161,8176\}$	
	$(10,1)$	1023	6	$\{1023,2046,4092,4607,6399,7295,7743,7967,8079,8135,8163,8177,8184\}$	
	$(11,1)$	2047	6	$\{2047,4094,5119,6655,7423,7807,7999,8095,8143,8167,8179,8185,8188\}$	
	$(4,2)$	85	8	$\{85,170,340,680,1360,2069,2565,2689,2720,4138,5130,5378,5440\}$	
	$(5,2)$	341	8	$\{341,682,1364,2133,2581,2693,2721,2728,4266,5162,5386,5442,5456\}$	

Table 4.2: Continued

Bibliography

[1] Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. *J. Cryptology*, 4:3–72, 1991.

[2] L. Budaghyan, C. Carlet, and A. Pott. New classes of almost bent and almost perfect nonlinear polynomials. *IEEE Transactions on Information Theory*, 52(3):1141–1152, 2006.

[3] Lilya Budaghyan, Nikolay S. Kaleyski, Soonhak Kwon, Constanza Riera, and Pantelimon Stănică. Partially APN Boolean functions and classes of functions that are not APN infinitely often. *Cryptography and Communications*, 12:527–545, 2020.

[4] Anne Canteaut, Alain Couvreur, and Léo Perrin. Recovering or Testing Extended-Affine Equivalence. *IEEE Transactions on Information Theory*, pages 1–1, 2022.

[5] Claude Carlet. *Boolean Functions for Cryptography and Coding Theory*. Cambridge University Press, 2021.

[6] Claude Carlet, Pascale Charpin, and Victor Zinoviev. Codes, Bent Functions and Permutations Suitable For DES-like Cryptosystems. *Des. Codes Cryptography*, 15:125–156, 11 1998.

[7] Florent Chabaud and Serge Vaudenay. Links between differential and linear cryptanalysis. In Alfredo De Santis, editor, *Advances in Cryptology — EUROCRYPT'94*, pages 356–365, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[8] Robert S. Coulter. On the evaluation of a class of Weil sums in characteristic 2. *New Zealand Journal of Mathematics*, 28(2):171–184, 1999.

[9] Itai Dinur and Adi Shamir. Breaking Grain-128 with Dynamic Cube Attacks. In Antoine Joux, editor, *Fast Software Encryption*, pages 167–187, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[10] H. Dobbertin. Almost Perfect Nonlinear Power Functions on $GF(2^n)$: The Welch Case. *IEEE Trans. Inf. Theor.*, 45(4):1271–1275, sep 2006.

[11] Hans Dobbertin. Almost Perfect Nonlinear Power Functions on $GF(2^n)$: The Niho Case. *Information and Computation*, 151(1):57–72, 1999.

[12] Hans Dobbertin. Almost Perfect Nonlinear Power Functions on $GF(2^n)$: A New Case for n Divisible by 5. In Dieter Jungnickel and Harald Niederreiter, editors, *Finite Fields and Applications*, pages 113–121, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[13] Yves Edel and Alexander Pott. On the Equivalence of Nonlinear Functions. In *Enhancing Cryptographic Primitives with Techniques from Error Correcting Codes*, 2009.

[14] Nikolay Kaleyski. Deciding EA-equivalence via invariants. *Cryptography and Communications*, 14(2):271–290, 2022.

[15] T. Kasami. The weight enumerators for several classes of subcodes of the 2nd order binary Reed-Muller codes. *Information and Control*, 18(4):369–394, 1971.

[16] Shuxing Li, Wilfried Meidl, Alexandr Polujan, Alexander Pott, Constanza Riera, and Pantelimon Stănică. Vanishing Flats: A Combinatorial Viewpoint on the Planarity of Functions and Their Application. *IEEE Transactions on Information Theory*, 66(11):7101–7112, 2020.

[17] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and Their Applications*. Cambridge University Press Cambridge; New York, 1986.

[18] Mitsuru Matsui. Linear Cryptanalysis Method for DES Cipher. In *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[19] Kaisa Nyberg. On the Construction of Highly Nonlinear Permutations. In *Advances in Cryptology - EUROCRYPT '92, Workshop on the Theory and Application of of Cryptographic Techniques, Balatonfüred, Hungary, May 24-28, 1992, Proceedings*, volume 658 of *Lecture Notes in Computer Science*, pages 92–98. Springer, 1992.

[20] Kaisa Nyberg. Differentially uniform mappings for cryptography. In Tor Helleseth, editor, *Advances in Cryptology — EUROCRYPT '93*, pages 55–64, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[21] Satoshi Yoshiara. Equivalences of quadratic APN functions. *Journal of Algebraic Combinatorics*, 35(3):461–475, 2012.

[22] Satoshi Yoshiara. Equivalences of power APN functions with power or quadratic APN functions. *Journal of Algebraic Combinatorics*, 44(3):561–585, 2016.