CONSTRUCTION OF SUBSTITUTION BOXES DEPENDING ON LINEAR BLOCK CODES

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF APPLIED MATHEMATICS

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

SENAY YILDIZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF CRYPTOGRAPHY

SEPTEMBER 2004

Approval of the Graduate School of Applied Mathematics

Prof. Dr. Aydın AYTUNA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ersan AKYILDIZ
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Melek D. YÜCEL
Supervisor

Examining Committee Members

Prof. Dr. Ersan AKYILDIZ

Prof. Dr. Kemal LEBLEBİCİOĞLU

Assoc. Prof. Dr. Ali DOĞANAKSOY

Assoc. Prof. Dr. Ferruh ÖZBUDAK

Assoc. Prof. Dr. Melek D. YÜCEL

Abstract

CONSTRUCTION OF SUBSTITUTION BOXES DEPENDING ON LINEAR BLOCK CODES

Yıldız, Senay
M.Sc., Department of Cryptography
Supervisor: Assoc. Prof. Dr. Melek D. YÜCEL

September 2004, 72 pages

The construction of a substitution box (S-box) with high nonlinearity and high resiliency is an important research area in cryptography.

In this thesis, t-resilient $n \times m$ S-box construction methods depending on linear block codes presented in "A Construction of Resilient Functions with High Nonlinearity" by T. Johansson and E. Pasalic in 2000, and two years later in "Linear Codes in Generalized Construction of Resilient Functions with Very High Nonlinearity" by E. Pasalic and S. Maitra are compared and the former one is observed to be more promising in terms of nonlinearity. The first construction method uses a set of nonintersecting $[n-d,\ m,\ t+1]$ linear block codes in deriving t-resilient S-boxes of nonlinearity $2^{n-1} - 2^{n-d-1}$, where d is a parameter to be maximized for high nonlinearity. For some cases, we have found better results than the results of Johansson and Pasalic, using their construction.

As a distinguished reference for $n \times n$ S-box construction methods, we study

the paper "Differentially Uniform Mappings for Cryptography" presented by K. Nyberg in Eurocrypt 1993. One of the two constructions of this paper, i.e., the inversion mapping described by Nyberg but first noticed in 1957 by L. Carlitz and S. Uchiyama, is used in the S-box of Rijndael, which is chosen as the Advanced Encryption Standard. We complete the details of some theorem and proposition proofs given by Nyberg.

Keywords: S-box, nonlinearity, resiliency, nonintersecting linear block codes, inversion mapping, differential uniformity.

Öz

DOĞRUSAL BLOK KODLAR KULLANARAK YERLEŞİM KUTULARININ OLUŞTURULMASI

Yıldız, Senay
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi: Doç. Dr. Melek D. YÜCEL

Eylül 2004, 72 sayfa

Kriptografide, eğriselliği ve esnekliği yüksek olan yerleşim kutularının oluşturulması önemli bir araştırma konusudur.

Bu tezde, T. Johansson ve E. Pasalic'in 2000'de yayınladığı, "A Construction of Resilient Functions with High Nonlinearity" ile E. Pasalic ve S. Maitra'nın iki yıl sonraki, "Linear Codes in Generalized Construction of Resilient Functions with Very High Nonlinearity " adlı makalelerinde tanıtılan, $n \times m$ boyutlu ve esnekliği t olan yerleşim kutusu tasarımları karşılaştırılarak, ilk yöntemin eğrisellik açısından daha iyi değerler verdiği gözlemlenmiştir. İlk tasarım, esnekliği t ve eğriselliği $2^{n-1}-2^{n-d-1}$ olan yerleşim kutularını oluşturmak için, $[n-d,\, m,\, t+1]$ parametreleri ve kesişmeyen doğrusal blok kodlar kullanmaktadır. Burada d, yerleşim kutusunun eğriselliğinin yüksek olması için mümkün olan en büyük değerde seçilmesi gereken bir parametredir. Çalışmalarımızda, bazı durumlar için Johansson ve Pasalic'in sonuçlarından daha iyileri, yine onların tasarımını kullanarak elde edilmiştir.

K. Nyberg'in, $n \times n$ yerleşim kutusu tasarımında çok tanınan, 1993 Euro-

crypt konferansında sunduğu“Differentially Uniform Mappings for Cryptography” adlı makalesini inceledik. Bu makaledeki iki yerleşim kutusu tasarımından biri olan, daha önce 1957 yılında L. Carlitz and S. Uchiyama'nın fark ettiği, bir cisim elemanının tersini alan fonksiyon, Gelişmiş Şifreleme Standardı olarak seçilen Rijndael algoritmasının yerleşim kutusunda da kullanılmıştır. Nyberg tarafından verilen bazı teorem ispatlarının ayrıntılarını tamamladık.

Anahtar Kelimeler: Yerleşim kutusu, eğrisellik, esneklik, kesişmeyen doğrusal blok kodlar, ters fonksiyon, türevsel düzenlilik.

to my parents,

Acknowledgments

I express sincere appreciation to my supervisor Assoc. Prof. Dr. Melek D. Yücel for her guidance, insight and cooperation throughout this study.

I am grateful to Assoc. Prof. Dr. Ali Doğanaksoy for guiding, encouraging and motivating me throughout my education at METU.

I want to thank my parents for their support and all the beautiful things that they have done for me.

I am also thankful to Çiğdem Özakın for her friendship that make me always confident. I am also thankful to her for her help in typing this study.

I am grateful to Dilek Ünal for her being always with me.

I want to thank my homemates for their support and help at home.

I am thankful to Selçuk Kavut for helping me in the programming part of this study.

I would like to thank to my managers and all my colleagues at RTB Eğitim Çözümleri for their support and patience.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Substitution boxes (or S-boxes) are vector Boolean functions, which are used frequently in cryptographic applications. Nonlinearity of an S-box must be high for the resistance of block ciphers against linear cryptanalysis [Meier & Staffelbach, 1989], [Heys, 2001], [Knudsen & Robshaw, 1994]. Resiliency is another important criterion in the design of S-boxes [Friedman, 1982], [Stinson, 1993], [Zhang & Zheng, 1997]. However, constructing S-boxes with high nonlinearity and resiliency is difficult [Stinson & Massey, 1995], [Cheon, 2001], [Kurosawa & Satoh & Yamamoto, 1997], since nonlinearity and resiliency are conflicting properties.

There are many S-box construction methods in the literature [Nyberg, 1993], [Nyberg, 1990], [Nyberg, 1992], [Webster & Tavares, 1985], [Kurosawa & Satoh & Yamamoto, 1997], [Johansson & Pasalic, 2000], [Pasalic & Maitra, 2002]. In this thesis, we study four of them. Two of these construction methods [Nyberg, 1993] are power polynomials defined in finite fields. The others [Johansson & Pasalic, 2000] and [Pasalic & Maitra, 2002] use the concept of linear block codes.

To provide the theoretical background for the thesis, some basic definitions related to the Boolean functions, S-boxes and coding theory are reviewed in Chapter 2. We then consider some properties of finite fields with particular emphasis on the trace function.

In Chapter 3, we summarize the two $n \times n$ S-box constructions given by

Nyberg in [Nyberg, 1993]. One of them is the inverse of the power polynomial $S(x) = x^{2^k+1}$ and the other is the inversion mapping $S(x) = x^{-1}$, both defined in $GF(2^n)$. We review the theorems and propositions used in these constructions following Nyberg, and whenever needed, we provide the details of the proofs to make them clearer. Both mappings have high nonlinearity, low differential uniformity, high algebraic degree and computational efficiency. These methods provide higher nonlinearity than the other methods we study in this thesis.

In Chapter 4, we review the construction method in [Johansson & Pasalic, 2000]. The method depends on finding a set of nonintersecting linear codes. The main problem in this construction method is finding the desired number of nonintersecting linear codes by a complete search. Moreover, there are some restrictions on the choice of n, m and t for t-resilient $n \times m$ S-boxes. We have implemented this method and for some values of n and m, we have found better results for nonlinearity than the results in [Johansson & Pasalic, 2000].

In Chapter 5, we summarize the construction method in [Pasalic & Maitra, 2002]. This construction is similar to the method in [Johansson & Pasalic, 2000] but it has the advantage of using only one linear code instead of using a set of nonintersecting linear codes for the construction of an $n \times m$ S-box. Each Boolean function is composed of a resilient function and a bent function. The important point is finding $2m$ different bent functions. There are also some restrictions on the choice of n, m and t for a t-resilient $n \times m$ S-box.

In Chapter 6, we present our programming results for the construction in [Johansson & Pasalic, 2000] and make a comparison with the results in [Johansson & Pasalic, 2000] and [Pasalic & Maitra, 2002].

Conclusions are discussed in Chapter 7.

Chapter 2

Theoretical Background

In this chapter, first we review some basic definitions related to the Boolean functions [Johansson & Pasalic, 2000], [Pasalic & Maitra, 2002] and [Siegenthaler, 1984]; S-boxes [Johansson & Pasalic, 2000], [Pasalic & Maitra, 2002] and [Nyberg, 1993] and coding theory [Blahut, 1983]. Then, we consider some properties of finite fields [Blahut, 1983] and [Lidl & Niederreiter, 1986] with particular emphasis on the trace function, which is an important concept to be used in the derivation of S-box properties.

2.1 Boolean Functions

Basic definitions and properties related with Boolean functions are stated below, following the definition of a field.

Definition 2.1.1. A *field* F is a set that has two operations defined on it; addition and multiplication, such that the following axioms are satisfied:

1. F is an abelian group under addition.
2. F is closed under multiplication and the set of nonzero elements is an abelian group under multiplication.
3. The distributive law $(a+b)c = ac + bc$ holds for all $a, b, c \in F$.

The field of real numbers ($\mathcal{R}$), the field of complex numbers ($\mathcal{C}$) and the field of rational numbers ($\mathcal{Q}$) are fields with infinite number of elements. A field with q elements, if it exists, is called a *finite field*, or a *Galois Field*, and is denoted by $GF(q)$ or F_q.

The smallest field is the field with elements 0 and 1. It is denoted by $GF(2)$. The addition and multiplication operations in $GF(2)$ are addition and multiplication in *mod* 2.

Definition 2.1.2. A *Boolean function* $f(x) : GF(2)^n \rightarrow GF(2)$ is the function which has the input all of the possible n tuples $x = (x_1, ..., x_n)$ of $GF(2)$ and produces an output of one bit. The set of all n-variable Boolean functions are denoted by V_n.

Definition 2.1.3. The *truth table* T_f of a Boolean function f is a 1×2^n vector defined as $T_f = [f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{2^n-1})]$, where α_i denotes the n-bit vector which corresponds to the binary representation of the integer $i = 0, 1, ..., 2^n - 1$.

Definition 2.1.4. The *sequence vector* S_f of a Boolean function f is a 1×2^n vector defined as $S_f = [(-1)^{f(\alpha_0)}, (-1)^{f(\alpha_1)}, ..., (-1)^{f(\alpha_{2^n-1})}]$, where α_i denotes the n-bit vector which corresponds to the binary representation of the integer $i = 0, 1, ..., 2^n - 1$.

Definition 2.1.5. A Boolean function f is called an *affine function* if it is in the form: $f(x) = b_1x_1 \oplus b_2x_2 \oplus \ldots \oplus b_nx_n \oplus c = w \cdot x \oplus c$, where $b_1, b_2, \ldots, b_n, c$ are elements of $GF(2)$, w and x are elements of $GF(2)^n$ and $\oplus$ and $\cdot$ denote addition and inner product operations in $GF(2)$. f is called *linear* if $c = 0$. The set of all n-variable affine Boolean functions are denoted by A_n and the set of all n-variable linear Boolean functions are denoted by L_n.

Definition 2.1.6. The *Hamming weight of a vector* $w \in GF(2)^n$ is the number of its nonzero components denoted by $wt(w)$.

Definition 2.1.7. The *Hamming weight of a Boolean function* f is the Hamming weight of its truth table T_f. Then, $wt(f) = wt(T_f)$ can be written.

Definition 2.1.8. The *Hamming distance* between two Boolean functions f and g is defined as $d(f,g) = wt(T_f \oplus T_g)$.

Since Boolean functions are the basic components of many cryptosystems, they have an important role in the design of cryptosystems. They should have some cryptographic properties such as high resiliency for stream ciphers and high nonlinearity for block ciphers. Let's look at some of these properties.

Definition 2.1.9. A Boolean function f is called *balanced* if $wt(f) = 2^{n-1}$, i.e., there must be equal number of 1's and 0's in the truth table of f.

Definition 2.1.10. The *nonlinearity* of a Boolean function $f : GF(2)^n \rightarrow GF(2)$ is the minimum Hamming distance of f from the set of all n-variable affine functions; to be denoted by $nl(f)$.

Definition 2.1.11. Let $f(x_1, ..., x_n)$ be an n-variable Boolean function. f can be represented as

$$f(x) = a_0 \oplus \bigoplus_{i=1}^{n} a_i x_i \oplus \bigoplus_{1 \leq i \neq j \leq n} a_{ij} x_i x_j \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n$$

where the coefficients $a_0, a_i, a_{ij}, \ldots, a_{12\ldots n} \in \{0, 1\}$. This representation is called the *algebraic normal form(ANF)* of f. The number of variables in the highest order product term with nonzero coefficient is called the *algebraic degree* or simply *degree* of f.

Definition 2.1.12. The *Walsh transform* of f is a real-valued function over $GF(2)^n$ defined as

$$W_f(w) = \sum_{x \in GF(2)^n} (-1)^{f(x)} (-1)^{x \cdot w} \tag{2.1}$$

where $w, x \in GF(2)^n$.

The nonlinearity criterion of Boolean functions can be quantified through the Walsh transform as follows:

Definition 2.1.13. Let $f(x_1, ..., x_n)$ be an n-variable Boolean function. Then

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{w \in GF(2)^n} |W_f(w)|.$$

Definition 2.1.14. Let $f(x_1, ..., x_n)$ be an n-variable Boolean function. If n is even and f has the maximum nonlinearity, then f is called a *bent* function. The nonlinearity of bent functions is $nl(f) = 2^{n-1} - 2^{\frac{n}{2}-1}$.

The following lemma was first proved by [Siegenthaler, 1984] and characterizes the correlation immunity in the Walsh transform domain.

Lemma 2.1.15. *A Boolean function f is m^{th} order correlation immune iff $W_f(w) = 0,\ w|1 \leq wt(w) \leq m,\ \forall w \in GF(2)^n$.*

Definition 2.1.16. An m^{th} order correlation immune Boolean function which is balanced is called an m^{th} *order resilient (m-resilient) function*, i.e., iff

$$W_f(w) = 0,\ w|wt(w) \leq m,\ \forall w \in GF(2)^n \tag{2.2}$$

[Siegenthaler, 1984] proved a fundamental relation between the number of variables n, degree d and order of correlation immunity m of a Boolean function: $m + d \leq n$. Moreover, if the function is balanced then $m + d \leq n - 1$.

Definition 2.1.17. Let f_1 and f_2 be $(n-1)$-variable Boolean functions. The concatenation of f_1 and f_2 is an n-variable function

$$f(x_1, \ldots, x_n) = (1 \oplus x_n) f_1(x_1, \ldots, x_{n-1}) \oplus x_n f_2(x_1, \ldots, x_{n-1})$$

and denoted by $f = f_1 || f_2$. The truth tables of f_1 and f_2 are concatenated to provide the truth table of the n-variable function defined above.

In cryptographic applications with stream ciphers, nonlinear Boolean functions are needed to combine linear feedback shift registers, while constructing keystream generators. These Boolean functions must satisfy certain properties,

i.e., balancedness, high nonlinearity, high algebraic degree and high resiliency, in order to increase the time/space complexity of the attacks such as Berlekamp-Massey linearity synthesis attack [Menezes, Oorschot, Vanstone, 1997] and different linear approximation attacks. Boolean functions are also used in block ciphers as component functions of the S-boxes. Now, let's look at the definition and some properties, which an S-box must satisfy.

2.2 Substitution Boxes

Basic definitions and properties related with S-boxes are stated below.

Definition 2.2.1. A function of the form $S : GF(2)^n \rightarrow GF(2)^m$ is called an $n \times m$ *S-box*, which takes n bits as the input and outputs m bits. If each output bit is called the n-variable Boolean function f_i, then $S(x) = (f_1(x), \ldots, f_m(x))$ where $x \in GF(2)^n$.

Usually, S-boxes are the only nonlinear parts of block ciphers. Below, we summarize some definitions related with the S-boxes.

Definition 2.2.2. The *nonlinearity of an S-box*, $S(x) = (f_1(x), \ldots, f_m(x))$, is the minimum of the nonlinearities of the Boolean functions formed by any linear combination of component functions, i.e.,

$$nl(S) = \min_f(nl(f)) = \min_f\{nl(f)|f = \bigoplus_{i=1}^{m} j_i f_i(x)\}$$

Definition 2.2.3. The *algebraic degree* of an S-box is the minimum of degrees of all nonzero linear combinations of the component functions of S, namely

$$deg(S) = \min_f deg\{f|f = \bigoplus_{i=1}^{m} j_i f_i(x)\}$$

Definition 2.2.4. An $n \times m$ S-box is t-resilient if and only if all nonzero linear combinations $\bigoplus_{i=1}^{m} j_i f_i(x)$ of $f_1(x), f_2(x), \ldots, f_m\ (x)$ are t-resilient.

Definition 2.2.5. Let $S = (f_1, f_2, ..., f_m)$ be a function from $GF(2)^n$ to $GF(2)^m$ where $1 \leq m \leq n$ and let $x = (x_1, x_2, ..., x_n) \in GF(2)^n$.

1. S is said to be *unbiased* with respect to a fixed subset $T = \{j_1, j_2, \ldots, j_t\}$ of $\{1, 2, \ldots, n\}$ if for every $(a_1, a_2, \ldots, a_t) \in GF(2)^t$,

$$(f_1(x), \ldots, f_m(x))|(x_{j_1} = a_1, \ldots, x_{j_t} = a_t)$$

   runs through all the vectors in $GF(2)^m$, each 2^{n-t-m} times when $(x_{i_1}, \ldots, x_{i_{n-t}})$ runs through $GF(2)^{n-t}$, where $t \geq 0$, $\{i_1, \ldots, i_{n-t}\} = \{1, ..., n\} - \{j_1, ..., j_t\}$ and $i_1 < i_2 \ldots < i_{n-t}$. In other words, if we fix any t bits of n-bit input, output runs through all the vectors in $GF(2)^m$ each 2^{n-t-m} times when the unfixed input runs through $GF(2)^{n-t}$.

2. S is a *t-resilient function* if S is unbiased with respect to every $T \subseteq \{1, ..., n\}$ with $|T| = t$. In other words, if all possible t bits in the input are fixed, the output will be uniform.

Lemma 2.2.6. *A function $S = (f_1, f_2, ..., f_m)$, where each f_i, $1 \leq i \leq m$, is a function from $GF(2)^n$ to $GF(2)$ is uniformly distributed (unbiased) iff all nonzero linear combinations of $f_1, f_2, \ldots, f_m$ are balanced.*

Definition 2.2.7. Let $GF(2)^n$ and $GF(2)^m$ be finite vector spaces. A mapping $S : GF(2)^n \to GF(2)^m$ is called *differentially δ-uniform* if for all $\alpha \in GF(2)^n$, $\alpha \neq 0$ and $\beta \in GF(2)^m$, $|\{z \in GF(2)^n | S(z + \alpha) + S(z) = \beta\}| \leq \delta$.

Proposition 2.2.8. *Let $A : GF(2)^n \to GF(2)^n$ and $B : GF(2)^m \to GF(2)^m$ be group isomorphisms and $S : GF(2)^n \to GF(2)^m$ be differentially δ-uniform. Then $B \circ S \circ A$ is differentially δ-uniform.*

Proposition 2.2.9. *Let $S : GF(2)^n \to GF(2)^n$ be a differentially δ-uniform bijection. Then the inverse mapping of S is also differentially δ-uniform.*

2.3 Linear Block Codes

In this section, there are some basic definitions and a useful theorem from coding theory [Blahut, 1983].

Definition 2.3.1. A *block code* of size M over an alphabet with q symbols is a set of M, q-ary sequences of length n called *codewords*. If $q = 2$, the symbols are called *bits*. Usually, $M = q^k$ for some integer k. Then the code is called (n, k) *code*.

Definition 2.3.2. A *linear block code* is a subspace of $GF(q)^n$. That is, a *linear code* is a nonempty set of n tuples over $GF(q)$ (codewords) such that the sum of two codewords is a codeword, and the product of any codeword by a field element is a codeword. Any set of basis vectors for the subspace can be used as rows to form a $k \times n$ matrix G, called the *generator matrix* of the code. Any codeword is a linear combination of the rows of G.

In any linear code, the all zero word, as the vector space origin, is always a codeword.

Definition 2.3.3. Let $C = \{c_i, i = 0, ..., M-1\}$ be a code. Then the *minimum distance* of C is the Hamming distance of the pair of codewords with smallest Hamming distance.

An (n, k) block code with minimum distance d^* is also described as an (n, k, d^*) block code.

Theorem 2.3.4. ***(Singleton Bound)*** *The minimum distance of any linear* (n, k, d^*) *code satisfies* $d^* \leq n - k + 1$.

2.4 Properties of Finite Fields

The following definitions, properties and proofs are taken from [Blahut, 1983] and [Lidl & Niederreiter, 1986].

Definition 2.4.1. If successively adding the multiplicative identity 1 to itself in $GF(q)$ never gives 0, then we say that $GF(q)$ has characteristic zero. Otherwise, there is a prime number p such that $1 + 1 + \ldots + 1$ (p times) equals 0, and p is called the *characteristic of the field* $GF(q)$. Then q is a power of p.

Definition 2.4.2. Let $GF(q)^*$ denote the set of nonzero elements of the finite field $GF(q)$. The *order* of $a \in GF(q)^*$ is the least positive integer k such that $a^k = 1$.

Definition 2.4.3. Let $GF(q)$ be a field. A subset of $GF(q)$ is called a *subfield* if it is a field under the inherited addition and multiplication. The original field is then called an *extension field* of the subfield.

Let's now look at some basic properties of Galois fields [Blahut, 1983] and [Lidl & Niederreiter, 1986]:

1. The order of any $a \in GF(q)^*$ divides $q - 1$.
2. If $GF(q)$ is a finite field with q elements, then every $a \in GF(q)$ satisfies $a^q = a$.
3. In any finite field, the number of elements is a power of a prime.
4. If p is prime and m is positive integer, the smallest subfield of $GF(p^m)$ is $GF(p)$.
5. In a finite field of characteristic 2, $-\beta = \beta$ for every β in the field.
6. If p is a prime and m is an integer, then there is a finite field with p^m elements.
7. If n divides m, $GF(p^n)$ is a subfield of $GF(p^m)$.
8. $(a + b)^p = a^p + b^p$ in any field of characteristic p, for every $a, b \in GF(q)$.

Definition 2.4.4. For $\alpha \in F = GF(q^m)$ and $K = GF(q)$, the *trace* $Tr_{F/K}(\alpha)$ *of* α *over* K is defined by $Tr_{F/K}(\alpha) = \alpha + \alpha^q + \ldots + \alpha^{q^{m-1}}$.

In particular, $Tr_{F/K}(\alpha)$ is always an element of $c \in K = GF(q)$.

Theorem 2.4.5. *Let $K = GF(q)$ and $F = GF(q^m)$. Then the trace function $Tr_{F/K}$ satisfies the following properties:*

1. $Tr_{F/K}(\alpha + \beta) = Tr_{F/K}(\alpha) + Tr_{F/K}(\beta)$ *for all* $\alpha, \beta \in GF(q^m)$.
2. $Tr_{F/K}(c\alpha) = cTr_{F/K}(\alpha)$ *for all* $c \in GF(q)$, $\alpha \in GF(q^m)$.
3. $Tr_{F/K}$ *is a linear transformation from* $GF(q^m)$ *onto* $GF(q)$, *where both* $GF(q^m)$ *and* $GF(q)$ *are viewed as vector spaces over* $GF(q)$.
4. $Tr_{F/K}(a) = ma$ *for all* $a \in GF(q)$.
5. $Tr_{F/K}(\alpha^q) = Tr_{F/K}(\alpha)$ *for all* $\alpha \in GF(q^m)$.

Proof:

1. For α, $\beta \in GF(q^m)$, use property 8 to get

$$\begin{aligned} Tr_{F/K}(\alpha+\beta) &= \alpha + \beta + (\alpha+\beta)^q + \ldots + (\alpha+\beta)^{q^{m-1}} \\ &= \alpha + \beta + \alpha^q + \beta^q + \ldots + \alpha^{q^{m-1}} + \beta^{q^{m-1}} \\ &= \alpha + \alpha^q + \ldots + \alpha^{q^{m-1}} + \beta + \beta^q + \ldots + \beta^{q^{m-1}} \\ &= Tr_{F/K}(\alpha) + Tr_{F/K}(\beta) \end{aligned}$$

2. For $c \in GF(q)$, we have $c^{q^j} = c$ for all $j \geq 1$ by property 2. Therefore, for $\alpha \in GF(q^m)$,

$$\begin{aligned} Tr_{F/K}(c\alpha) &= c\alpha + c^q\alpha^q + \ldots + c^{q^{m-1}}\alpha^{q^{m-1}} \\ &= c\alpha + c\alpha^q + \ldots + c\alpha^{q^{m-1}} \\ &= c(\alpha + \alpha^q + \ldots + \alpha^{q^{m-1}}) \\ &= cTr_{F/K}(\alpha) \end{aligned}$$

3. By parts 1 and 2, $Tr_{F/K}(\alpha)$ is always an element of K. Moreover, the properties 1 and 2 show that $Tr_{F/K}$ is a linear transformation from $GF(q^m)$ into K. To show that this mapping is onto, it is enough to show that $\exists \alpha \in GF(q^m)$ with $Tr_{F/K}(a) \neq 0$. Now, $Tr_{F/K}(a) = 0$ iff a is a root of the polynomial $x^{q^{m-1}} + \ldots + x^q + x \in K[x]$ in $GF(q^m)$. This polynomial can have at most q^{m-1} roots in $GF(q^m)$ and $GF(q^m)$ has q^m elements. So, a cannot be a root of this polynomial.

4. By definition of the trace function and property 2,

$$\begin{aligned} Tr_{F/K}(a) &= a + a^q + \ldots + a^{q^{m-1}} \\ &= a + a + \ldots + a \ (m \ times) \\ &= ma \end{aligned}$$

5. For $\alpha \in GF(q^m)$, $\alpha^{q^m} = \alpha$ by property 2, and so

$$\begin{aligned} Tr_{F/K}(\alpha^q) &= \alpha^q + \ldots + \alpha^{q^m} \\ &= Tr_{F/K}(\alpha) \end{aligned}$$

□

Theorem 2.4.6. *Let $F = GF(q^m)$ be a finite extension of the finite field $K = GF(q)$, both considered as vector spaces over $GF(q)$. Then the linear transformations from $GF(q^m)$ into $GF(q)$ are exactly the mappings L_β, $\beta \in GF(q^m)$, where $L_\beta(\alpha) = Tr_{F/K}(\beta\alpha)$ for all $\alpha \in GF(q^m)$. Furthermore, $L_\beta \neq L_\gamma$ whenever β and γ are distinct elements of $GF(q^m)$.*

Proof: Each mapping L_β is a linear transformation from $GF(q^m)$ into $GF(q)$

by Theorem 2.4.5 item 3. For $\beta, \gamma \in F$ with $\beta \neq \gamma$, we have

$$\begin{aligned} L_\beta(\alpha) - L_\gamma(\alpha) &= Tr_{F/K}(\beta\alpha) - Tr_{F/K}(\gamma\alpha) \\ &= Tr_{F/K}((\beta - \gamma)\alpha) \\ &\neq 0 \end{aligned}$$

for suitable $\alpha \in GF(q^m)$ since $Tr_{F/K}$ maps $GF(q^m)$ onto $GF(q)$, and so the mappings L_β and L_γ are different. The mappings L_β yield q^m different linear transformations from $GF(q^m)$ into $GF(q)$. On the other hand, every linear transformation from $GF(q^m)$ into $GF(q)$ can be obtained by assigning arbitrary elements of $GF(q)$ to the m elements of a given basis of $GF(q^m)$ over $GF(q)$. Since this can be done in q^m different ways, the mappings L_β already exhaust all possible linear transformations from $GF(q^m)$ into $GF(q)$. □

Chapter 3

Differentially Uniform Mappings

S-box construction is one of the most important topics in cryptography. Being the only nonlinear part of a block cipher, the S-box should be constructed such that, its nonlinearity is as high as possible. There are many construction methods in the literature: [Nyberg, 1993], [Nyberg, 1990], [Nyberg, 1992], [Webster & Tavares, 1985], [Kurosawa & Satoh & Yamamoto, 1997], [Johansson & Pasalic, 2000], [Pasalic & Maitra, 2002].

[Nyberg, 1993] gives two examples of transformations of $GF(2^n)$ with high nonlinearity, high nonlinear order and efficient construction and computability, which we review in this chapter. The first one is the power polynomial $S(x) = x^{2^k+1}$ and its inverse. The second transformation is the inversion mapping $S(x) = x^{-1}$ which is used in the Advanced Encryption Standard (Rijndael).

We complete the details of the Nyberg's proofs in Propositions 3.1.1 and 3.2.1.

3.1 Power Polynomials $S(x) = x^{2^k+1}$ in $GF(2^n)$ and Their Inverses

In this section, we review the properties of the power polynomial $S(x) = x^{2^k+1}$ and its inverse [Nyberg, 1993].

Proposition 3.1.1. *Let $S(x) = x^{2^k+1}$ be a power polynomial in $GF(2^n)$ and let $s = gcd(k, n)$. Then S is differentially 2^s uniform. If $\frac{n}{s}$ is odd, that is, S is a permutation, then the Hamming distance of the Boolean function $f_w(x) = tr(wS(x))$ from the set of linear Boolean functions is equal to $2^{n-1} - 2^{\frac{n+s}{2}-1}$, for all $w \in GF(2^n), w \neq 0$.*

Proof: If the number of x vectors satisfying the inequality, $S(x+\alpha)+S(x) < \beta$ is 2^s or less for given $\alpha, \beta \in GF(2^n)$, then $S(x) = x^{2^k+1}$ is differentially 2^s uniform. One needs to show that

$$(x+\alpha)^{2^k+1} + x^{2^k+1} = \beta \tag{3.1}$$

has 2^s solutions or less, where s=$gcd(k, n)$. Equation (3.1) has either zero or at least two solutions since:

$$\begin{aligned}
(x+\alpha)^{2^k+1} + x^{2^k+1} &= \beta \\
(x+\alpha)^{2^k}(x+\alpha) + x^{2^k+1} &= \beta \\
(x^{2^k} + \alpha^{2^k})(x+\alpha) + x^{2^k+1} &= \beta \\
x^{2^k+1} + x^{2^k}\alpha + \alpha^{2^k}x + \alpha^{2^k+1} + x^{2^k+1} &= \beta \\
x^{2^k}\alpha + \alpha^{2^k}x + (\alpha^{2^k+1} + \beta) &= 0
\end{aligned}$$

If $k = 1$, this is a 2^{nd} degree equation and it has 2 solutions. Now, assume x_1 and x_2 are two different solutions. Then

$$(x_1+\alpha)^{2^k+1} + x_1^{2^k+1} = \beta \tag{3.2}$$

$$(x_2+\alpha)^{2^k+1}+x_2^{2^k+1}=\beta \tag{3.3}$$

Adding equations (3.2) and (3.3),

$(x_1+\alpha)^{2^k+1}+x_1^{2^k+1}+(x_2+\alpha)^{2^k+1}+x_2^{2^k+1}=0$

$(x_1^{2^k}+\alpha^{2^k})(x_1+\alpha)+x_1^{2^k+1}+(x_2^{2^k}+\alpha^{2^k})(x_2+\alpha)+x_2^{2^k+1}=0$

$x_1^{2^k+1}+\alpha x_1^{2^k}+\alpha^{2^k}x_1+\alpha^{2^k+1}+x_1^{2^k+1}+x_2^{2^k+1}+\alpha x_2^{2^k}+\alpha^{2^k}x_2+\alpha^{2^k+1}+x_2^{2^k+1}=0$

$\alpha x_1^{2^k}+\alpha x_2^{2^k}+\alpha^{2^k}x_1+\alpha^{2^k}x_2=0$

$(x_1^{2^k}+x_2^{2^k})\alpha+(x_1+x_2)\alpha^{2^k}=0$

$(x_1+x_2)^{2^k}\alpha+(x_1+x_2)\alpha^{2^k}=0 \quad (divide\ by\ \alpha(x_1+x_2))$ (3.4)

$(x_1+x_2)^{2^k-1}+\alpha^{2^k-1}=0$

$(x_1+x_2)^{2^k-1}=\alpha^{2^k-1}$

Since $gcd(n,k)=s$, $\exists c\in Z^+$ s.t. $n=cs \Rightarrow 2^s-1$ divides 2^n-1. Also, $\exists c'\in Z^+$ s.t. $k=c's \Rightarrow 2^s-1$ divides 2^k-1. Then one can write $2^k-1=(2^s-1)c''$. If $n=cs$, then there exists a subfield G with 2^s elements s.t. $\forall g\in G$, $g^{2^s-1}=1$. Now, $x_1+x_2=\alpha$ is a solution. Then $x_1+x_2=\alpha g$ is also a solution since $g^{2^k-1}=(g^{2^s-1})^{c''}=1$. Then one can write $(x_1+x_2)^{2^k-1}=(g\alpha)^{2^k-1}=g^{2^k-1}\alpha^{2^k-1}=\alpha^{2^k-1}\Rightarrow x_1+x_2\in\alpha(G\backslash\{0\})$.

Secondly, to compute the nonlinearity, as $nl(f_w(x))=2^{n-1}-2^{\frac{n+s}{2}-1}$, it is enough to show $\max\limits_{t\in GF(2^n)}|W_{f_w}(t)|=2^{\frac{n+s}{2}}$ using the Walsh transform defined by equation (2.1). Let $t\in GF(2^n)$. Then

$$\begin{aligned}(W_{f_w}(t))^2 &= \sum_{x\in GF(2^n)}(-1)^{f_w(x)+t\cdot x}\sum_{y\in GF(2^n)}(-1)^{f_w(x+y)+t\cdot(x+y)}\\ &= \sum_{y\in GF(2^n)}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{f_w(x+y)+f_w(x)}\end{aligned} \tag{3.5}$$

Let $y \neq 0$. Consider the linear mapping

$$\phi : x \rightarrow S(x+y) + S(x) + S(y) \tag{3.6}$$

$$\begin{aligned} S(x+y) + S(x) + S(y) &= (x+y)^{2^k+1} + x^{2^k+1} + y^{2^k+1} \\ &= (x+y)^{2^k}(x+y) + x^{2^k+1} + y^{2^k+1} \\ &= (x^{2^k} + y^{2^k})(x+y) + x^{2^k+1} + y^{2^k+1} \\ &= x^{2^k+1} + x^{2^k}y + xy^{2^k} + y^{2^k+1} + x^{2^k+1} + y^{2^k+1} \\ &= x^{2^k}y + xy^{2^k} \end{aligned}$$

Let E_y denote the range of the mapping (3.6), and find its kernel.

$$x \in Ker(\phi) \Rightarrow x^{2^k}y + y^{2^k}x = 0 \tag{3.7}$$

Since equation (3.7) is similar to equation (3.4), the kernel of this linear mapping is yG. The dimension of yG is s, the dimension of $GF(2^n)$is n, so the dimension of the linear space E_y is $n - s$. Note that the trace function

$$tr(w\beta) : E_y \rightarrow GF(2) \tag{3.8}$$

maps each $\beta \in E_y$ to an element of $GF(2)$. If the trace function is onto, then

$$\dim(E_y) = \dim(\ker(tr(w\beta))) + \dim(\mathrm{Im}(tr(w\beta)))$$

Then $n - s = \dim(\ker(tr(w\beta))) + 1$, and so $\dim(\ker(tr(w\beta))) = n - s - 1$. This means that the function $tr(w\beta)$ takes the value 0 for 2^{n-s-1} times and the value 1 for 2^{n-s-1} times. Then $\sum_{\beta \in E_y} (-1)^{tr(w\beta)} = 0$. Consider the case when the trace function is not onto. The trace function takes the value 0 at least once. But, since this function is not onto, the trace function must always take the value 0. Then $tr(w\beta)) = 0$. Then it can be concluded that, for each $y \neq 0$, either $tr(w\beta) = 0$ for all $\beta \in E_y$ or $\sum_{\beta \in E_y} (-1)^{tr(w\beta)} = 0$. The vectors y which gives

$tr(w\beta)) = 0$ for all $\beta \in E_y$ form a linear subspace Y, i.e.,

$$Y = y|tr(w\beta) = 0, \forall \beta \in E_y \tag{3.9}$$

$f_w(x+y) + f_w(x) + f_w(y) = tr(w(x^{2^k}y + y^{2^k}x)) = 0$. Now, if we use these ideas in equation (3.5), we get the equation

$$\begin{aligned}(W_{f_w}(t))^2 &= \sum_{y\in Y}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{f_w(x+y)+f_w(x)} \\ &\quad +\sum_{y\notin Y}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{tr(w(x^{2^k}y+y^{2^k}x))+f_w(y)}\end{aligned}$$

When $y \in Y$, $f_w(x+y) + f_w(x) = f_w(y)$. When $y \notin Y$, $f_w(x+y) + f_w(x) = tr(w(x^{2^k}y + y^{2^k}x)) + f_w(y)$. Then,

$$\begin{aligned}(W_{f_w}(t))^2 &= \sum_{y\in Y}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{f_w(x+y)+f_w(x)} \\ &\quad +\sum_{y\notin Y}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{tr(w(x^{2^k}y+y^{2^k}x))+f_w(y)} \\ &= \sum_{y\in Y}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{f_w(y)} \\ &\quad +\sum_{y\notin Y}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{tr(w(x^{2^k}y+y^{2^k}x))+f_w(y)} \\ &= \sum_{y\in Y}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{f_w(y)} \\ &\quad +\sum_{y\notin Y}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{tr(w(x^{2^k}y+y^{2^k}x))}(-1)^{f_w(y)} \\ &= \sum_{y\in Y}(-1)^{t\cdot y}\sum_{x\in GF(2^n)}(-1)^{f_w(y)} \\ &\quad +\sum_{y\notin Y}(-1)^{t\cdot y}(-1)^{f_w(y)}\sum_{x\in GF(2^n)}(-1)^{tr(w(x^{2^k}y+y^{2^k}x))}\end{aligned}$$

Note that, $\sum_{x\in GF(2^n)}(-1)^{tr(w(x^{2^k}y+y^{2^k}x))} = 0$ by considering the map

$$\upsilon : GF(2^n) \to GF(2)$$

$$x \rightarrow tr(w(x^{2^k}y + y^{2^k}x)) \tag{3.10}$$

The map (3.10) is linear and onto so half of the values of the function must be 1 and half of them is 0. Then $\sum_{x \in GF(2^n)} (-1)^{tr(w(x^{2^k}y + y^{2^k}x))} = 0$. Combining this,

$$\begin{aligned}
(W_{f_w}(t))^2 &= \sum_{y \in Y}(-1)^{t \cdot y} \sum_{x \in GF(2^n)} (-1)^{f_w(y)} \\
&= \sum_{y=0}(-1)^{t \cdot y} \sum_{x \in GF(2^n)} (-1)^{f_w(y)} + \sum_{y \in Y \setminus \{0\}} (-1)^{t \cdot y} \sum_{x \in GF(2^n)} (-1)^{f_w(y)} \\
&= 2^n + 2^n \sum_{y \in Y \setminus \{0\}} (-1)^{t \cdot y + f_w(y)}
\end{aligned}$$

By definition of Y by equation (3.10), the function f_w is linear on Y. Hence it remains to show that Y has 2^s elements. Since f_w is linear on Y, then

$$W_{f_w}(t) = \sum_{x \in GF(2^n)} (-1)^{f_w(x) + t \cdot x} = 2^n \delta(t + t_0)$$

$W_{f_w}(t)$ takes the value 2^n for $t = t_0$. Otherwise the value of the function is 0. Then one can write

$$W_{f_w}(t_0) = \sum_{x \in GF(2^n)} (-1)^{f_w(x) + t_0 \cdot x} = 2^n$$

Then $(-1)^{f_w(x) + t_0 \cdot x} = 1$, so $f_w(x) + t_0 \cdot x = 0$. Since $Y \subset GF(2^n)$, we can write

$$W_{f_w}(t) = \sum_{y \in Y \setminus \{0\}} (-1)^{f_w(t) + t \cdot y} = 2^{s-1} \delta(t + t_0)$$

$f_w(x) + t_0 \cdot x = 0 \Rightarrow W_{f_w}(t) = 2^{s-1}$. Then we get the conclusion. Let $y \in Y$. Then $tr(wyx^{2^k}) = tr(wy^{2^k}x) = tr(w^{2^k}y^{2^{2k}}x^{2^k})$ for all $x \in GF(2^n)$. (This equation is due to the linearity of f_w and the properties of the trace function.)

Then,

$$\begin{aligned} tr(wyx^{2^k}) &= tr(w^{2^k}y^{2^{2k}}x^{2^k}) \\ wy &= w^{2^k}y^{2^{2k}} \\ \frac{wy}{wy} &= \frac{w^{2^k}y^{2^{2k}}}{wy}(y \neq 0) \\ 1 &= w^{2^k-1}y^{2^{2k}-1} \end{aligned}$$

Note that $y^{2^{2k}-1} = \frac{(y^{2^k})^2}{y} = y^{2^k-1}y^{2^k}$. Then;

$$1 = w^{2^k-1}y^{2^k-1}$$

$$1 = (wS(y))^{2^k-1}(S(y) = y^{2^k+1}) \tag{3.11}$$

From equation (3.11), 2^s-1 nonzero solutions y can be found, since S is assumed to be a permutation. This completes the proof.

□

Remark 3.1.2. One can observe the following arguments due to Proposition 3.1.1:

1. $\forall x \in GF(2^n)$, the resulting Boolean function $f_w(x) = tr(wS(x))$ is one of the linear combination of component functions of $S(x) : GF(2^n) \rightarrow GF(2^n)$.

2. If n is odd, $1 < k < n$ and $gcd(n, k) = 1$, then the power polynomial $S(x) = x^{2^k+1}$ in $GF(2^n)$ is a differentially 2-uniform permutation.

3. If $n = 2^m$ for some $m \in Z^+$, then $S(x) = x^{2^k+1}$ in $GF(2^n)$ is never a permutation. If $\frac{n}{s}$ is odd, S is permutation.

Let $w_2(k)$ denote the 2-weight of a non-negative integer k. The following proposition is well-known and stated with proof in [Carlet, 1990].

Proposition 3.1.3. *Let $w \in GF(2^n)$, $w \neq 0$ and let $x \rightarrow x^e$ be a permutation of $GF(2^n)$. Then $deg(tr(wx^e)) = w_2(e)$.*

The permutations $x \rightarrow x^{2^k+1}$ in $GF(2^n)$, n odd, are highly nonlinear, resistant against the differential cryptanalysis and have efficient construction and computability but their output functions are only quadratic as stated in [Nyberg, 1993]. Their inverses, however, have degrees linearly growing with n [Nyberg, 1993].

Proposition 3.1.4. *Let n be odd, $gcd(n,k) = 1$ and $S(x) = x^{2^k+1}$. Then*

$$S^{-1}(x) = x^l,$$

where

$$l = \frac{2^{k(n+1)}}{2^{2k}-1} = \sum_{i=0}^{\frac{n-1}{2}} 2^{2ik} \mod(2^n - 1)$$

with $w_2(l) = \frac{n+1}{2}$.

Proof:

$$\begin{aligned} l(2^k+1) &= \sum_{i=0}^{\frac{n-1}{2}} 2^{(2i+1)k} + \sum_{i=0}^{\frac{n-1}{2}} 2^{2ik} \mod(2^n-1) \\ &= \sum_{i=0}^{n} 2^{ik} \mod(2^n-1) \\ &= \sum_{i=0}^{n} 2^{i} \mod(2^n-1) \qquad (3.12) \\ &= 2^{n+1} - 1 \mod(2^n-1) \\ &= 1 \mod(2^n-1) \end{aligned}$$

where the equality (3.12) follows from the fact that the mapping $i \rightarrow ki$ permutes the integers modulo n if $gcd(n,k) = 1$.

□

As a conclusion of this section, the following properties of the inverse of

$S(x) = x^{2^k+1}$ in $GF(2^n)$ with n odd and $gcd(n, k) = 1$ can be listed:

1. $nl(S^{-1}) = \min_{w\neq 0} \min_{L\in L_n} \min_{x\in GF(2^n)} d(tr(wS^{-1}(x)), L(x)) = 2^{n-1} - 2^{\frac{n-1}{2}}$.

2. $deg(tr(wS^{-1}(x))) = w_2((2^k + 1)^{-1} \mod(2^n - 1)) = \frac{n+1}{2}$.

3. S^{-1} is differentially 2-uniform.

4. Using the fast exponentiation algorithm, the computation of $S^{-1}(x)$ is of poynomial time requiring $\frac{n-1}{2}$ squarings and $\frac{n-1}{2}$ multiplications in $GF(2^n)$.

3.2 The Mapping $S(x) = x^{-1}$ in a Finite Field

The inverse mapping $S : GF(2^n) \to GF(2^n)$ is first noticed in 1957 by Carlitz and Uchiyama [Carlitz & Uchiyama, 1957] and defined [Nyberg, 1993] by $S(x) = x^{-1}$ if $x \neq 0$ and $S(x) = 0$ if $x = 0$. The differential uniformity and nonlinearity properties of inversion mapping are as follows:

Proposition 3.2.1. *The inversion mapping is differentially* 4*- uniform if* n *is even and differentially* 2*-uniform if* n *is odd. Moreover,*

$$nl(S) = \min_{w\neq 0} \min_{L\in L_n} \min_{x\in GF(2^n)} d(tr(wx^{-1}),\ L(x)) \geq 2^{n-1} - 2^{\frac{n}{2}}.$$

Proof: Let $\alpha,\ \beta \in GF(2^n)$ and $\alpha \neq 0$ and look at the solutions of the following equation:

$$(x + \alpha)^{-1} - x^{-1} = \beta \tag{3.13}$$

It is enough to show that, the equation (3.13) has a solution set consists of at most 2 elements if n is odd and consists of 4 elements if n is even. Assume that

$x \neq 0$ and $x \neq -\alpha$. Then multiplying equation (3.13) by $x(x+\alpha)$;

$$\begin{aligned} x(x+\alpha)[(x+\alpha)^{-1} - x^{-1}] &= \beta x(x+\alpha) \\ x - (x+\alpha) &= \beta x(x+\alpha) \\ -\alpha &= \beta x^2 + \alpha\beta x \end{aligned}$$

$$\beta x^2 + \alpha\beta x + \alpha = 0 \tag{3.14}$$

Equation (3.14) has at most two solutions in $GF(2^n)$. If either $x = 0$ or $x = -\alpha$ is solution to (3.13), then both of them are solutions and $\beta = \alpha^{-1}$. Then (3.14) is equivalent to

$$\beta\beta^{-1}x^2 + \alpha\beta\beta^{-1}x + \alpha\beta^{-1} = 0 \ \text{ (multiply equation (3.14) by } \beta^{-1})$$

$$x^2 + \alpha x + \alpha^2 = 0 \ \ (\beta = \alpha^{-1} \Rightarrow \beta^{-1} = \alpha) \tag{3.15}$$

Equation (3.15) gives 2 more solutions to $(x+\alpha)^{-1} - x^{-1} = \beta$. Let's solve equation (3.14) in the special case in $GF(2^n)$. By squaring (3.14) and substituting $x^2 = \alpha x + \alpha^2$,

$$\begin{aligned} (x^2 + \alpha x + \alpha^2)^2 &= x^4 + \alpha^2 x^2 + \alpha^4 \\ &= x^4 + \alpha^2(\alpha x + \alpha^2) + \alpha^4 \\ &= x^4 + \alpha^3 x + \alpha^4 + \alpha^4 \\ &= x(x^3 + \alpha^3) \end{aligned}$$

If $gcd(3, 2^n - 1) = 1$ or equivalently n is odd, $x(x^3 + \alpha^3)$ has no other solutions than $x = 0$ or $x = \alpha$. If n is even, 3 divides $2^n - 1$. Let $d = \frac{1}{3}(2^n - 1)$. Then there are two more solutions which are $x = \alpha^{1+d}$ and $x = \alpha^{1+2d}$. Then S is differentially 2-uniform if n is odd and differentially 4-uniform if n is even.

Now, to show $nl(S) \geq 2^{n-1} - 2^{\frac{n}{2}}$, it is enough to show $max|W_S(t)| \leq 2^{\frac{n}{2}+1}$, i.e.,

$max|W_S^2(t)| \leq 2^{n+2}$ as in the equation

$$W_S^2(t) = 2^n + 2^n \sum_{y \in Y \setminus \{0\}} (-1)^{ty+f_w(y)}$$

where Y is the subspace which is defined similarly as in the first section.

If n is odd, $G = \{0, 1\}$, i.e., has only two elements. Then

$$\begin{aligned} W_S^2(t) &= 2^n + 2^n \sum_{y \in Y \setminus \{0\}} (-1)^{ty+f_w(y)} \\ &= 2^n + 2^n \times 1 \\ &= 2^{n+1} \end{aligned}$$

Then $W_S^2(t) \leq 2^{n+2}$.

If n is even, $G = \{0, 1, \alpha^d, \alpha^{2d}\}$, i.e., G has four elements. Then

$$\begin{aligned} W_S^2(t) &= 2^n + 2^n \sum_{y \in Y \setminus \{0\}} (-1)^{ty+f_w(y)} \\ &= 2^n + 2^n \times 3 \\ &= 2^{n+2} \end{aligned}$$

Then $W_S^2(t) \leq 2^{n+2}$. □

Remark 3.2.2. As a conclusion of this section, the following properties of the inverse mapping can be listed:

1. $nl(S) = \min_{w \neq 0} \min_{L \in L_n} \min_{x \in GF(2^n)} d(tr(wS(x)), L(x)) \geq 2^{n-1} - 2^{\frac{n}{2}}$.

2. $deg(tr(wx^{-1})) = w_2(2^n - 2) = n - 1$.

3. S is differentially 2-uniform if n is odd and it is differentially 4-uniform if n is even.

4. The Euclidean algorithm computes x^{-1} in polynomial time with respect to n.

Chapter 4

A Construction Of Resilient Functions With High Nonlinearity

In this chapter, the $n \times m$ S-box (where $m < n$) construction method introduced in [Johansson & Pasalic, 2000] is described in Sections 4.1 and 4.2. Section 4.3 gives the lower bounds on the cardinality of a set of linear nonintersecting codes, which is an essential part of the mentioned construction. We generate a simple example and present it in Section 4.4.

To show the restrictions on design parameters, we provide Table 4.1, which shows the highest possible nonlinearity values achievable by this method for $n \times m$ S-boxes with $n = 6$, 7 and 8. Later in Chapter 6, we present our construction results for larger values of n, using the Johansson & Pasalic method described in the following sections.

4.1 Construction of the Function

The construction in [Johansson & Pasalic, 2000] depends on finding a set of nonintersecting linear codes. For the construction of $n \times m$ and t-resilient S-box with nonlinearity $2^{n-1} - 2^{n-d-1}$, the number of needed nonintersecting codes

is $\left\lceil \frac{2^d}{2^m-1} \right\rceil$, where d is a parameter which needs to be maximized in order to get high nonlinearity. The construction method is based on the following theorem [Johansson & Pasalic, 2000].

Theorem 4.1.1. *Let n, m, t and d be four positive integers with $n \geq 4$, $1 \leq t \leq n-3$, $1 \leq d \leq n-t$, $m \leq n-d$. For each pair (y,i), where $y \in GF(2)^d$, $i = 1, \ldots, m$, let $A_y^i \in GF(2)^{n-d}$ s.t. $wt(A_y^i) \geq t+1$.*

For $a \in GF(2)^{n-d}$, $c = (c_1, \ldots, c_m) \in GF(2)^m$, let

$$s_{a,c}^* = |\{y \in GF(2)^d | \sum_{i=1}^{m} c_i A_y^i = a\}| \text{ and } s^* = \max_{c \in GF(2)^m} \max_{a \in GF(2)^{n-d}} s_{a,c}^*.$$

The S-box $S : GF(2)^n \to GF(2)^m$ is constructed by $S(y,x) = (A_y^1 x, \ldots, A_y^m x)$ where each $A_y^i x \in GF(2)$,

$$y = (y_1, y_2, ..., y_d) \in GF(2)^d \text{ and } x = (x_1, x_2, ..., x_{n-d}) \in GF(2)^{n-d}.$$

Then the followings hold:

1. *S is uniformly distributed if $\sum_{i=1}^{m} c_i A_y^i \neq 0$ for any $c \in GF(2)^m$, $c \neq 0$.*
2. *S is t-resilient if for any $a \in GF(2)^{n-d}$ s.t. $0 \leq wt(a) \leq t$ and $c \in GF(2)^m$, $c \neq 0$, it holds that $\sum_{i=1}^{m} c_i A_y^i \neq a$.*
3. *$nl(S) = 2^{n-1} - s^* 2^{n-d-1}$.*

Proof: Let $g_c : GF(2)^n \to GF(2)$ be the Boolean function defined by

$$g_c(y,x) = \sum_{i=1}^{m} c_i A_y^i x$$

for $c \in GF(2)^m$, $c \neq 0$.

1. Since S is of the form $S(y,x) = (A_y^1 x, \ldots, A_y^m x)$, g_c is any linear combination of component functions of S.

Then,

$$W_{g_c}(0) = \sum_{y,x}(-1)^{g_c(y,x)} = \sum_{y}\sum_{x}(-1)^{(c_1A_y^1+...+c_mA_y^m)x} = 0.$$

By assumption $\sum_{i=1}^{m} c_i A_y^i \neq 0$ for any $c \in GF(2)^m$, $c \neq 0$.

$$W_{g_c}(0) = 0 \Rightarrow g_c \text{ is balanced.}$$

Since g_c denotes the any linear combination of component functions of S, S is uniformly distributed by Lemma 2.2.6.

2. It is enough to show all nonzero linear combinations of f_1, f_2, ..., f_m are t-resilient. In order to see that, $g_c(y,x)$ is t-resilient, one needs to show that the Walsh transform $W_{g_c}(b,a)$ defined by equation (2.1) is zero according to equation (2.2) for all $(b,a) \in GF(2)^n$ with $wt(b,a) \leq t$ and $a, x \in GF(2)^{n-d}$ and $b, y \in GF(2)^d$

$$\begin{aligned} W_{g_c}(b,a) &= \sum_{(y,x)\in GF(2)^n} (-1)^{g_c(y,x)}(-1)^{(b,a)\cdot(y,x)} \\ &= \sum_{(y,x)\in GF(2)^n} (-1)^{g_c(y,x)}(-1)^{b\cdot y \oplus a\cdot x} \\ &= \sum_{y}(-1)^{b\cdot y}\sum_{x}(-1)^{(c_1A_y^1+...+c_mA_y^m+a)\cdot x} \end{aligned}$$

Since $0 \leq wt(a) \leq t$, $\sum_{i=1}^{m} c_i A_y^i \neq a$. Then $\sum_{x}(-1)^{(c_1A_y^1+...+c_mA_y^m+a)\cdot x} = 0$.
So,

$$\begin{aligned} W_{g_c}(b,a) &= \sum_{y}(-1)^{b\cdot y} \times 0 \\ &= 0 \end{aligned}$$

Then, $g_c(y,x)$ is t-resilient.

3. As in the previous part, to obtain the nonlinearity of S, one finds the nonlinearity of $g_c(y,x)$, since the nonlinearity of $S : GF(2)^n \to GF(2)^m$ is the minimum nonlinearity of all linear combinations of component functions.

   By the previous part,

$$\begin{aligned} W_{g_c}(b,a) &= \sum_y (-1)^{b\cdot y} \sum_x (-1)^{(c_1 A_y^1 + \ldots + c_m A_y^m + a)\cdot x} \\ &= 2^{n-d} \sum_{\{y \mid \sum_{i=1}^m c_i A_y^i = a\} = s^*_{a,c}} (-1)^{b\cdot y} \end{aligned}$$

   Hence,

$$\max |W_{g_c}(b,a)| \le 2^{n-d} \max_{c \in GF(2)^m} \max_a s^*_{a,c} = s^* 2^{n-d}$$

   If $b = 0$ in $2^{n-d} \sum_{\{y \mid \sum_{i=1}^m c_i A_y^i = a\} = s^*_{a,c}} (-1)^{b\cdot y}$, then

$$|W_{g_c}(0,a)| = 2^{n-d} |\{y | \sum_{i=1}^m c_i A_y^i = a\}| = 2^{n-d} s^*_{a,c}.$$

   It follows that

$$\begin{aligned} \max |W_{g_c}(b,a)| &\ge \max |W_{g_c}(0,a)| \\ &= 2^{n-d} \max_{c \in GF(2)^m} \max_a s^*_{a,c} \\ &= s^* 2^{n-d} \end{aligned}$$

   Therefore,

$$\max_{b,a} |W_{g_c}(b,a)| = s^* 2^{n-d}.$$

   Then,

$$nl(S) = 2^{n-1} - s^* 2^{n-d-1}.$$

□

In this construction, the component functions are actually a concatenation of 2^d linear t-resilient functions in $n-d$ variables as stated in [Johansson & Pasalic, 2000]. Thus, $y \in GF(2)^d$ can be viewed as a specific address to some

linear function. It can be obtained a large number of distinct functions by permuting the values of $(A_y^1, \ldots, A_y^m)$ with same parameters.

Let the matrix A be formed by entries A_y^i's.

$$A = \begin{bmatrix} A_{0\ldots0}^1 & A_{0\ldots0}^2 & . & . & . & A_{0\ldots0}^m \\ A_{0\ldots1}^1 & A_{0\ldots1}^2 & . & . & . & A_{0\ldots1}^m \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ A_{1\ldots1}^1 & A_{1\ldots1}^2 & . & . & . & A_{1\ldots1}^m \end{bmatrix}_{2^d \times m} \tag{4.1}$$

Since the nonlinearity of S is $nl(S) = 2^{n-1} - s^* 2^{n-d-1}$, it depends on s^* and d where s^* is the maximum number of identical vectors appearing in any linear combination of A's columns. In this construction $s^* = 1$. So the value of d should be maximized in order to get high nonlinearity. If S is t-resilient, the vectors contained in each row of the matrix A spans an $[n-d, m, t+1]$ linear code. If one wants to achieve $s^* = 1$, then

$$\sum_{i=1}^{m} c_i A_y^i \neq \sum_{i=1}^{m} c_i A_{y'}^i, \forall c = (c_1, \ldots, c_m) \neq 0, \text{ if } y \neq y'.$$

4.2 How to Construct the Matrix A

The nonlinearity of S depends on the value of d. For high nonlinearity, d must be maximized. For any component function $A_y^i x$ of S, the vectors in $GF(2^{n-d})$ should be chosen with weight greater than the order of resiliency. Then one gets the inequality

$$\binom{n-d}{t+1} + \binom{n-d}{t+2} + \ldots + \binom{n-d}{n-d} \geq 2^d \tag{4.2}$$

where the left hand side is the number of $n-d$ bit A_y^i's with Hamming weight at least $t+1$ and the right hand side is the number of y's.

When constructing the matrix A, Lemma 4.2.1 in [Johansson & Pasalic, 2000] is used.

Lemma 4.2.1. *Let $c_0, \ldots, c_{m-1}$ be a basis of a binary $[n-d, m, t+1]$ linear code C. Let β be a primitive element in $GF(2^m)$ and $(1, \beta, \beta^2, \ldots, \beta^{m-1})$ be a polynomial basis of $GF(2^m)$. Define a bijection $\phi : GF(2^m) \rightarrow C$ by*

$$\phi(a_0 + a_1\beta + \ldots + a_{m-1}\beta^{m-1}) = a_0c_0 + a_1c_1 + \ldots + a_{m-1}c_{m-1}$$

Consider the matrix

$$A^* = \begin{bmatrix} \phi(1) & \phi(\beta) & . & . & . & \phi(\beta^{m-1}) \\ \phi(\beta) & \phi(\beta^2) & . & . & . & \phi(\beta^m) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \phi(\beta^{2^m-2}) & \phi(1) & . & . & . & \phi(\beta^{m-2}) \end{bmatrix}_{2^m-1 \times m} \tag{4.3}$$

In any linear combination of columns (not all zero) of the matrix A^*, each nonzero codeword of C will appear exactly once.

Proof: Since ϕ is a bijection, it is enough to show that the matrix

$$D = \begin{bmatrix} 1 & \beta & . & . & . & \beta^{m-1} \\ \beta & \beta^2 & . & . & . & \beta^m \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \beta^{2^m-2} & 1 & . & . & . & \beta^{m-2} \end{bmatrix}_{2^m-1 \times m}$$

has the property that each element in $GF(2^m)^*$ will appear once in any combination of columns of the above matrix.

Any nonzero linear combination of columns can be written as

$$(c_0 + c_1\beta + \ldots + c_{m-1}\beta^{m-1}) \begin{bmatrix} 1 \\ \beta \\ \vdots \\ \beta^{2^m-2} \end{bmatrix}$$

for some $c_0, c_1, \ldots, c_{m-1} \in GF(2)$ and the statement is obvious. □

The following conclusion is in [Johansson & Pasalic, 2000]. In fact, it is obvious after Lemma 4.2.1.

Conclusion 4.2.2. To construct the $2^d \times m$ matrix A given by (4.1), the $(2^m - 1) \times m$ matrix A^*in (4.3) can fill the $2^m - 1$ row of A. Each nonzero codeword will then appear exactly once in each column and row. Then one selects another $[n - d, m, t + 1]$ linear code, constructs another $A^*$and fills the $2^m - 1$ rows. Totally $\lceil \frac{2^d}{2^m-1} \rceil$ nonintersecting linear codes are needed.

This result is stated as a theorem in [Johansson & Pasalic, 2000].

Theorem 4.2.3. *If there exists a set of linear $[n - d, m, t + 1]$ nonintersecting linear codes with cardinality $\lceil \frac{2^d}{2^m-1} \rceil$, then there exists a t-resilient function S : $GF(2^n) \to GF(2^m)$ with nonlinearity $nl(S) = 2^{n-1} - 2^{n-d-1}$.*

4.3 Lower Bounds on the Cardinality of a Set of Linear Nonintersecting Codes

In this section, two lower (existence) bounds on the cardinality of a set of nonintersecting linear codes are given [Johansson & Pasalic, 2000].

Lemma 4.3.1. *Let $GF(2)^n$ be an n-dimensional vector space over $GF(2)$ and $0 \leq k \leq m \leq n$. Let $N(n, m)$ denote the number of m-dimensional vector subspaces of $GF(2)^n$. Furthermore, let $N'(n, m, k)$ denote the number of m-dimensional vector subspaces containing a given k-dimensional vector subspace*

of $GF(2)^n$. Then the following is valid:

$$N(n,m) = \frac{\prod_{i=n-m+1}^{n}(2^i-1)}{\prod_{i=1}^{m}(2^i-1)} \text{ and } N'(n,m,k) = N(n-k,m-k) \tag{4.4}$$

Proof: The reader is referred to [Wan, 1993]. □

Let $M(n,m,d_{\min})$ denote the maximal cardinality of a set of nonintersecting linear codes for any given code parameters $n, m, d_{\min}$. Using Lemma 4.3.1, the following existence bound on $M(n,m,d_{\min})$ can be obtained [Johansson & Pasalic, 2000].

Theorem 4.3.2. *Let the codes in the set have parameters $[n,m,d_{\min}]$ and let $D = \{x \in GF(2^n) | 1 \leq wt(x) \leq d_{\min}-1\}$. Then $M(n,m,d_{\min})$ is lower bounded by*

$$M(n,m,d_{\min}) \geq \left\lceil \frac{N(n,m) - |D|N(n-1,m-1)}{(2^m-1)(N(n-1,m-1)-1)} \right\rceil$$

Proof: Since the minimum distance of all the codes is $d_{\min}$, none of them is allowed to intersect the sphere D. Let $\mathbf{C}$ denote the set of all linear codes of length n and dimension m. According to the previous lemma, the total number of codes is $N(n,m)$. Any element(vector) in D is a 1-dimensional vector space. The number of codes containing an arbitrary word $x \in D$ is $N(n-1,m-1)$. Removing all codes in $\mathbf{C}$ intersecting an element in D, i.e., all codes having too low minimum distance leaves us with at least $N(n,m) - N(n-1,m-1)|D|$ codes in $\mathbf{C}$.

In general, some codes will contain more than one codeword from D and hence $N(n,m) - N(n-1,m-1)|D|$ is an upper bound on the number of codes intersecting the sphere D.

Now, one can choose any code, say C', of the remaining codes in $\mathbf{C}$. An upper bound on the number of codes intersecting C' in more than the zero

word is now derived.

$$|\{C \in \mathbf{C} | C \cap C' \neq \{0\}\}| \leq (2^m - 1)(N(n-1, m-1) - 1)$$

This inequality is a consequence of the simple fact that any of $2^m - 1$ nonzero codewords of $C\,'$ can be in at most $N(n-1, m-1)$ codes.

One continues to select a new code C'' and removes all codes that intersect C'', etc. It then follows that an M^{th} code can be added to the set of nonintersecting codes if the following inequality holds,

$$N(n,m) - |D|N(n-1,m-1) - (M-1)(2^m - 1)(N(n-1,m-1) - 1) \geq 0.$$

From this one gets the bound. □

A second lower bound on the cardinality of a set of nonintersecting linear codes is obtained by considering the set of all possible permutations on the codewords (i.e. column permutations) for a given linear code C. Thus, the condition for this lower bound is the existence of a linear $[n, m, d_{\min}]$ code C together with its weight distribution. Once a such code is known, one can compute a lower bound on $M(n, m, d_{\min})$ which will depend on the weight distribution [Johansson & Pasalic, 2000].

Theorem 4.3.3. *(*Permutation Bound) *Let C be a given $[n, m, d_{\min}]$ linear code specified by its weight distribution $T(D) = \sum_{i=d_{\min}}^{n} w_i D^i$. Then*

$$M(n, m, d_{\min}) \geq \left\lfloor \frac{n!}{\sum_{i=d_{\min}}^{n} w_i^2 i!(n-i)!} \right\rfloor$$

Proof: Let $A = \{1, 2, \ldots, n\}$ and let $S_n = \{\pi : A \to A\}$ be a set of all permutations on n letters acting on C with cardinality $n!$. Furthermore, let $C^{w_i} = \{c \in C : w_H(c) = i\}$ be a set of cardinality $|C^{w_i}| = w_i$. If $\prod^{w_i}$ is the set of all permutations that map any codeword in C^{w_i} to some codeword contained

in C^{w_i}, i.e., $\prod^{w_i} = \{\pi \in S_n : \pi(c) \in C^{w_i}, \text{ for some } c \in C^{w_i}\}$. Then

$$|\prod^{w_i}| = w_i^2 i!(n-i)!.$$

The idea is to remove all permutations π which map any nonzero codeword of C into C. Thus, the number of permutations to be discarded in order to obtain a code $\pi(C)$ which does not intersect C in more than the zero word is given by $\sum_{i=d_{\min}}^{n} w_i^2 i!(n-i)!$ and the condition for a second code will be $n! > \sum_{i=d_{\min}}^{n} w_i^2 i!(n-i)!$. Clearly one can proceed in the same manner, discarding all permutations which maps any nonzero codeword of C into $\pi(C)$, as long as having remaining permutations.

Thus, the M^{th} code can be added provided

$$n! - (M-1)\sum_{i=d_{\min}}^{n} w_i^2 i!(n-i)! \geq 0.$$

Rearranging the last inequality, the bound is obtained. □

4.4 Our Example

In the following, we show the details of the construction method in [Johansson & Pasalic, 2000] for the design of a 4×2 S-box of nonlinearity 4 and resiliency 1.

Example 4.4.1. Let's construct an S-box $S(y, x) : GF(2)^4 \rightarrow GF(2)^2$.

$n = 4$, $m = 2$, $y \in GF(2)$ and $x \in GF(2)^3$.

1. Let's choose $t = 1 (1 \leq t \leq n-3)$.
2. Choose $d = 2$ and check whether the inequality $\binom{2}{2} \geq 2^2$ holds or not. Since $1 \ngeq 4$ then $d \neq 2$. Then decrease d by one.

3. Choose $d = 1$ and check whether the inequality $\binom{3}{2} + \binom{3}{3} \geq 2$ holds or not. Since $4 \geq 2$ we can take $d = 1$.

4. We need $\left\lceil \frac{2^d}{2^m - 1} \right\rceil = \left\lceil \frac{2}{3} \right\rceil = 1$ linear $[n - d, m, t + 1] = [3, 2, 2]$ code. Let this code be $C = \{110, 011, 101, 000\}$. Notice that each nonzero codeword in C can be chosen as an entry of the matrix $A = \begin{bmatrix} A_o^1 & A_o^2 \\ A_1^1 & A_1^2 \end{bmatrix}$ by using equation (4.1). Remembering that every nonzero codeword appears only once in each row and column of A, we can take $A = \begin{bmatrix} 110 & 011 \\ 011 & 101 \end{bmatrix}$.

5. The S-box is $S(y, x) = (A_y^1 \cdot x,\ A_y^2 \cdot x)$.

   Note that $wt(A_y^i) \geq 2$ makes the component functions 1-resilient. The 4×2 S-box constructed above finds the 2-bit output corresponding to the 4-bit input $(y, x) = (y, x_1, x_2, x_3)$ as follows:

   $y = 0 \Rightarrow$ we take A_0^1 and A_0^2.

$$\begin{aligned} S(y,x) &= S(y, x_1, x_2, x_3) = S(0, x_1, x_2, x_3) = (A_0^1 \cdot x, A_0^2 \cdot x) \\ &= ((110) \cdot (x_1, x_2, x_3), (011) \cdot (x_1, x_2, x_3)) \\ &= (x_1 + x_2, x_2 + x_3) \end{aligned}$$

   $y = 1 \Rightarrow$ we take A_1^1 and A_1^2.

$$\begin{aligned} S(y,x) &= S(y, x_1, x_2, x_3) = S(1, x_1, x_2, x_3) = (A_1^1 \cdot x, A_1^2 \cdot x) \\ &= ((011) \cdot (x_1, x_2, x_3), (101) \cdot (x_1, x_2, x_3)) \\ &= (x_2 + x_3, x_1 + x_3) \end{aligned}$$

   Then $S(y, x) = \left\{ \begin{array}{ll} (x_1 + x_2, x_2 + x_3) & \text{if } y = 0 \\ (x_2 + x_3, x_1 + x_3) & \text{if } y = 1 \end{array} \right\}$

$$nl(S) = 2^{n-1} - 2^{n-d-1} = 4$$

For a more complicated construction example of an 10×4 S-box, the reader is referred to [Johansson & Pasalic, 2000]. Finally, to give an idea about the restrictions on the parameters of this construction, we generate Table 4.1.

It should be remembered that the parameters of the construction [Johansson & Pasalic, 2000] are restricted as: $n \geq 4$, $1 \leq t \leq n-3$, $1 \leq d \leq n-t$ and $m \leq n-d$. Moreover, $t \leq n-d-m$ must be satisfied. So, Table 4.1 shows the highest nonlinearity values achievable by this construction for $n \times m$ S-boxes for $n = 6$, 7 and 8, choosing the maximum possible value of d for the associated values of n, m and t.

Table 4.1: Highest Possible Nonlinearity and d Values, $(nl(S)/d_{\max})$ of the Johansson & Pasalic Construction for $n \times m$ S-boxes

	1-resilient			2-resilient			3-resilient		
m	$n=6$	$n=7$	$n=8$	$n=6$	$n=7$	$n=8$	$n=6$	$n=7$	$n=8$
2	24/2	56/3	112/3	24/2	48/2	112/3	16/1	48/2	96/2
3	24/2	56/3	112/3	16/1	48/2	112/3	–	32/1	96/2
4	16/1	48/2	112/3	–	32/1	96/2	–	–	64/1
5	–	32/1	96/2	–	–	64/1	–	–	–
6	–	–	64/1	–	–	–	–	–	–

Chapter 5

Linear Codes In Generalized Construction Of Resilient Functions With Very High Nonlinearity

The construction presented by E. Pasalic and S. Maitra provides a method to construct highly nonlinear t-resilient functions $S : GF(2)^n \rightarrow GF(2)^m$ [Pasalic & Maitra, 2002]. The construction is based on the use of linear codes together with highly nonlinear multiple output functions and summarized in this chapter.

The construction takes a linear $[u, m, t+1]$ code and constructs highly nonlinear, t-resilient $n \times m$ S-box for $n > u$.

Section 5.1 provides the preliminary information, Section 5.2 reviews the construction method, and Section 5.3 presents the improvements of the construction, all are summarized from [Pasalic & Maitra, 2002]. In Section 5.4, we present our example of a 1-resilient 13×3 S-box constructed by the mentioned method.

5.1 Preliminaries

Basic definitions and properties related with the construction are stated below.

Let C denote the binary linear $[u, m, t+1]$ code with a set of basis vectors $c_0, c_1, \ldots, c_{m-1}$. The construction [Pasalic & Maitra, 2002] uses Lemma 4.2.1. Let A^* denote the matrix which is constructed by means of Lemma 4.2.1. There are $2^m - 1$ rows in the matrix A^*. Let $a_{i,j}$ denote the element in the i^{th} row and j^{th} column of A^*, for $i = 0, \ldots, m-1$. The corresponding linear function $a_{i,j}(x) = x \cdot a_{i,j}$ is t-resilient. According to Lemma 4.2.1, any column of the matrix A^* can be seen as a column vector of $2^m - 1$ distinct t-resilient linear functions on u-variables.

The construction of Pasalic and Maitra [Pasalic & Maitra, 2002] is similar to the construction of Johansson and Pasalic [Johansson & Pasalic, 2000]. But in the construction of Pasalic and Maitra [Pasalic & Maitra, 2002], there is no need to search for nonintersecting linear codes. It is only taken a single linear code with given parameters and use a repetition of the codewords in a specific manner.

Taking 2^q rows of A^* for $0 \leq q \leq m-1$ and denoting this matrix by D, the entries of the matrix D are of the form $d_{i,j} = a_{i,j}$, $i = 0, \ldots, 2^q - 1$ and $j = 0, \ldots, m-1$, i.e., $d_{i,j} = \phi(\beta^{i+j}) \in GF(2)^u$.

By Lemma 4.2.1, in any linear combination of columns of the matrix D, each nonzero codeword of C will either appear exactly once or not appear at all. Let the set $\{g_1, \ldots, g_m\}$ of Boolean functions on $(u+q)$-variables be defined as:

$g_{j+1}(y, x) = \bigoplus_{\eta \in GF(2)^q} (y_1 + \eta_1) \ldots (y_q + \eta_q)(d_{[\eta],j} \cdot x)$ where $[\eta]$ denotes the integer representation of vector η and $j = 0, \ldots, m-1$. To the j^{th} column of D, one associates the function g_{j+1}. From definition of $g_{j+1}(y, x)$, for g to be nonzero, y and η must be complement to each other.

The following proposition states the resiliency of linear combinations of the

functions g_j [Pasalic & Maitra, 2002].

Proposition 5.1.1. *Any nonzero linear combination of the functions $g_1, \ldots, g_m$ is a t-resilient function.*

Proof: Let $g(y,x) = \bigoplus_{i=1}^{m} \eta_j g_j$ for some $\eta \in GF^*(2)^m$. It is needed to show that $W_g(w) = 0$ for any w with $wt(w) \leq t$.

Then, by using equation (2.1), for any $(b,a) \in GF(2)^q \times GF(2)^u$ with $wt(b,a) \leq t$,

$$\begin{aligned} W_g(b,a) &= \sum_{y,x} (-1)^{g(y,x) \oplus (b,a)\cdot(y,x)} \\ &= \sum_{y,x} (-1)^{g(y,x)} (-1)^{(b,a)\cdot(y,x)} \\ &= \sum_{y} (-1)^{b\cdot y} \sum_{x} (-1)^{g(y,x) \oplus x\cdot a} \end{aligned}$$

For any fixed y, by Lemma 4.2.1, the function

$$g(y,x) = x \cdot \bigoplus_{j=0}^{m-1} c_j d_{[\eta],j}$$

is a linear function. Here, $c_j = \eta_{j+1}$ for $j = 0, \ldots, m-1$. Now, $wt(b,a) \leq t$ implies that $wt(a) \leq t$ and consequently, the right hand sum is zero which completes the proof.

□

The following proposition states the nonlinearity of linear combinations of the functions g_j [Pasalic & Maitra, 2002].

Proposition 5.1.2. *Any nonzero linear combination of the functions $g_1, \ldots, g_m$ has nonlinearity $2^{u+q-1} - 2^{u-1}$.*

Proof: $nl(g_j) = 2^{u+q-1} - 2^{u-1}$ for $j = 1, \ldots, m$ [Sarkar & Maitra, 2000]. Moreover, from Lemma 4.2.1, it is clear that any nonzero linear combination of these functions $g_1, \ldots, g_m$ will have the same property. □

The Corollary 5.1.3 and 5.1.4 and the proof of Corollary 5.1.4 are in [Pasalic & Maitra, 2002].

Corollary 5.1.3. *Given a* $[u, m, t+1]$ *linear code, it is possible to construct* $(u+q, m, t, 2^{u+q-1} - 2^{u-1})$ *functions for* $0 \leq q \leq m-1$.

Corollary 5.1.4. *It is possible to construct an* $(n' = 2m,\ m,\ 1,\ 2^{n'-1} - 2^{\frac{n'}{2}})$ *function* $S(x)$.

Proof: There exists an $[m+1, m, 2]$ linear code. Put $u = m+1$ and $q = m-1$, then

$$\begin{aligned} u+q &= m+1+m-1 \\ &= 2m \\ &= n' \\ u+q-1 &= n'-1 \\ t+1 &= 2 \\ \Rightarrow t &= 1 \\ u-1 &= m \\ &= \frac{n\prime}{2} \end{aligned}$$

Then $(n', m, 1, 2^{n'-1} - 2^{\frac{n'}{2}})$ functions exist.

□

The following proposition is well-known [Sarkar & Maitra, 2000].

Proposition 5.1.5. *Let* $h(y) \in V_k$ *and* $g(x) \in V_{n_1}$. *The nonlinearity of* $f(y, x) = h(y) \oplus g(x)$ *is given by*

$$nl(f) = 2^k nl(g) + 2^{n_1} nl(h) - 2nl(g)nl(h)$$

The Corollaries 5.1.6, 5.1.7 and 5.1.8 [Pasalic & Maitra, 2002] are simple consequences of Proposition 5.1.5.

Corollary 5.1.6. *Let $h(y)$ be a bent function on V_k, $k = 2m$. Let $g(x) \in V_{n_1}$ with $nl(g) = 2^{n_1-1} - 2^{u-1}$, for $u \leq n_1$. Then, the nonlinearity of $f(y,x) = h(y) \oplus g(x)$ is given by*

$$nl(f) = 2^{n_1+k-1} - 2^{\frac{k}{2}} 2^{u-1}.$$

Proof: Since $h(y)$ is a bent function on V_k, $k = 2m$, then

$$nl(h) = 2^{k-1} - 2^{\frac{k}{2}-1} \text{ and } nl(g) = 2^{n_1-1} - 2^{u-1}$$

is given. Then using Proposition 5.1.5;

$$\begin{aligned} nl(f) &= 2^k(2^{n_1-1} - 2^{u-1}) + 2^{n_1}(2^{k-1} - 2^{\frac{k}{2}-1}) - 2(2^{n_1-1} - 2^{u-1})(2^{k-1} - 2^{\frac{k}{2}-1}) \\ &= 2^{k+n_1-1} - 2^{k+u-1} + 2^{n_1+k-1} - 2^{n_1+\frac{k}{2}-1} - (2^{n_1} - 2^u)(2^{k-1} - 2^{\frac{k}{2}-1}) \\ &= 2^{k+n_1-1} - 2^{k+u-1} + 2^{n_1+k-1} - 2^{n_1+\frac{k}{2}-1} - 2^{n_1+k-1} + 2^{n_1+\frac{k}{2}-1} + 2^{u+k-1} - 2^{u+\frac{k}{2}-1} \\ &= 2^{k+n_1-1} - 2^{u+\frac{k}{2}-1} \end{aligned}$$

Then

$$nl(f) = 2^{n_1+k-1} - 2^{\frac{k}{2}} 2^{u-1}.$$

□

Corollary 5.1.7. *Let $h'(y')$ be a bent function on $V_k, k = 2r$ and let $h(y)$ be a function on V_{k+1} given by $h(y) = x_{k+1} \oplus h'(y')$. Let $g(x) \in V_{n_1}$ with $nl(g) = 2^{n_1-1} - 2^{u-1}$ for $u \leq n_1$. Then the nonlinearity of $f(y,x) = h(y) \oplus g(x)$ is given by $nl(f) = 2^{n_1+k-1} - 2^{\frac{k+1}{2}} 2^{u-1}$.*

Proof: $h(y) = x_{k+1} \oplus h'(y')$ where $x_{k+1} \in V_1$ and $h'(y') \in V_k$.

$$nl(h) = 2nl(h') + 2^k nl(x_{k+1}) - 2nl(h')nl(x_{k+1})$$

Since $nl(x_{k+1}) = 0$, $nl(h) = 2nl(h')$. Then

$$\begin{aligned} nl(h) &= 2(2^{k-1} - 2^{\frac{k}{2}-1}) \\ &= 2^k - 2^{\frac{k}{2}}. \end{aligned}$$

Now, $nl(h) = 2^k - 2^{\frac{k}{2}}$ is known and $nl(g)=2^{n_1-1} - 2^{u-1}$ is given. Then,

$$\begin{aligned} nl(f) &= 2^{k+1}(2^{n_1-1} - 2^{u-1}) + 2^{n_1}(2^k - 2^{\frac{k}{2}}) - (2^{n_1-1} - 2^{u-1})(2^k - 2^{\frac{k}{2}})2 \\ &= 2^{n_1+k} - 2^{u+k} + 2^{n_1+k} - 2^{n_1+\frac{k}{2}} - (2^{n_1} - 2^u)(2^k - 2^{\frac{k}{2}}) \\ &= 2^{n_1+k} - 2^{u+k} + 2^{n_1+k} - 2^{n_1+\frac{k}{2}} - 2^{n_1+k} + 2^{n_1+\frac{k}{2}} + 2^{u+k} - 2^{u+\frac{k}{2}} \\ &= 2^{n_1+k} - 2^{u+\frac{k}{2}} \end{aligned}$$

□

Corollary 5.1.8. *Let $h(y)$ be a constant function on V_k, $k > 0$. Let $g(x) \in V_{n_1}$ with $nl(g) = 2^{n_1-1} - 2^{u-1}$, for $u \leq n_1$. Then the nonlinearity of $f(y, x) = h(y) \oplus g(x)$ is given by $nl(f) = 2^{n_1+k-1} - 2^k 2^{u-1}$.*

Proof: Since $h(y)$ is a constant function, then $nl(h) = 0$. Using Proposition 5.1.5,

$$\begin{aligned} nl(f) &= 2^k(2^{n_1-1} - 2^{u-1}) \\ &= 2^{n_1+k-1} - 2^k 2^{u-1} \end{aligned}$$

□

Now, by using Corollary 5.1.6, Corollary 5.1.7 and the function g_j which is constructed before, one can construct highly nonlinear, resilient Boolean functions on higher number of variables using the composition of bent functions $h(y)$ with resilient functions g_j.

In order to use the same technique for the construction of an S-box, one needs to find a binary vector space of bent functions of dimension m.

Let A be of size $2^m \times m$ matrix given by $A = (\frac{0}{A^*})$ where A^* is a matrix constructed by means of Lemma 4.2.1 using $c_0, c_1, \ldots, c_{m-1}$ that spans an $[m, m, 1]$ code C with the unitary matrix I as the generator matrix.

Now, $A = (\frac{0}{A^*})$. Since A has 2^m columns and each entry is a codeword of length m, if one multiplies each entry with the vector $(x_1, x_2, \ldots, x_m)$, linear

functions on m variables are obtained. Then each column of the matrix A can be seen as a concatenation of 2^m distinct linear functions on m variables.

Also using Lemma 4.2.1, it is clear that any nonzero linear combination of these bent functions will provide a bent function. The algebraic degree of this class of bent functions is equal to m. So, the following proposition is proved [Pasalic & Maitra, 2002].

Proposition 5.1.9. *It is possible to obtain a binary vector space of bent functions on $2m$ variables of dimension m. Also,*

$$deg(\bigoplus_{i=1}^{m} \eta_i b_i) = m$$

where $b_1, \ldots, b_m$ is the basis and $\eta \in GF^*(2)^m$.

Proposition 5.1.10. *It is possible to obtain m distinct bent functions on $2p$-variables ($p \geq m$), say $b_1, \ldots, b_m$ such that any nonzero linear combination of these bent functions will provide a bent function. Also,*

$$deg(\bigoplus_{i=1}^{m} \eta_i b_i) = p$$

for $\eta \in GF^*(2)^m$.

5.2 Construction

In this section, we summarize the construction method of Pasalic and Maitra [Pasalic & Maitra, 2002].

Previously, the matrix A^* which has entries from a linear $[u, m, t+1]$ code C is constructed. Then the first 2^q rows of A^* for $0 \leq q \leq m-1$ are used to form the matrix D. For each column of D, one constructs the functions $g_1, \ldots, g_m$ which are $(u+q)$-variable functions with order of resiliency t and nonlinearity $2^{u+q-1} - 2^{u-1}$. Any nonzero linear combination of these functions will provide a

$(u+q)$-variable function g with order of resiliency t and nonlinearity $2^{u+q-1} - 2^{u-1}$.

In fact, it is desired to construct n-variable functions. It is clear that the $(u+q)$-variable function needs to be repeated 2^{n-u-q} times to make an n-variable function. An $(n-u-q)$-variable function is xored with the $(u+q)$-variable function to get an n-variable function. For the maximum possible nonlinearity by this method, the $(n-u-q)$-variable function must be of maximum possible nonlinearity. One uses m different functions $h_1, \ldots, h_m$ and use the compositions $f_1 = h_1 \oplus g_1, \ldots, f_m = h_m \oplus g_m$ to get m different n-variable functions. Then, any nonzero linear combination of $f_1, \ldots, f_m$ can be seen as the xor of linear combinations of $h_1, \ldots, h_m$ and linear combinations of $g_1, \ldots, g_m$. In fact this is the method of P. Sarkar and S. Maitra [Sarkar & Maitra, 2000]. In order to get high nonlinearity of the vector output function, high nonlinearity of the functions $h_1, \ldots, h_m$ and also high nonlinearity for their linear combinations are needed.

If $n-u-q$ is even, one can use bent functions $h_1, \ldots, h_m$. It is important that, m different bent functions as in Proposition 5.1.9 are needed such that the nonzero linear combinations will also produce bent functions. For this, the condition $n-u-q \geq 2m$ must be satisfied as in Proposition 5.1.10.

If $n-u-q$ is odd, one can use bent functions b_j of $(n-u-q-1)$-variables and take $h_j = x_n \oplus b_j$. This requires the condition $n-u-q-1 \geq 2m$ to get m distinct bent functions as in Proposition 5.1.10.

The value of $n-u-q$ may be less than $2m$ and it is not possible to get $2m$ bent functions.

The resiliency is satisfied by the functions g_i's and nonlinearity is satisfied by the functions h_i's. Since when calculating the nonlinearity of an S-box one takes all linear combinations, the linear combinations of g_i's should be high resilient and h_i's should be highly nonlinear.

Theorem 5.2.1 states the nonlinearity of an S-box constructed by this method

[Pasalic & Maitra, 2002].

Theorem 5.2.1. *Given a linear* $[u, m, t+1]$ *code, it is possible to construct an* $(n, m, t, nl(S))$ *function where* $S = (f_1, \ldots, f_m)$, $\pi = n - u - m + 1$ *and*

$$nl(S) = \left\{ \begin{array}{l} 2^{n-1} - 2^{u-1}, u \leq n < u+m \quad (1) \\ 2^{n-1} - 2^{n-m}, u+m \leq n < u+2m \quad (2) \\ 2^{n-1} - 2^{u+m-1}, u+2m \leq n < u+3m \quad (3) \\ 2^{n-1} - 2^{\frac{n+u-m-1}{2}}, n \geq u+3m-1, \pi \ even \quad (4) \\ 2^{n-1} - 2^{\frac{n+u-m}{2}}, n \geq u+3m, \ \pi \ odd \quad (5) \end{array} \right\}$$

Proof: In the proof, the functions $g_1, \ldots, g_m$ on $(u+q)$-variables which are basically concatenations of q distinct linear functions on u-variables will be used. From Proposition 5.1.2. for any $\tau \in GF(2)^m$,

$$nl(\bigoplus_{j=1}^{m} \tau_j g_j) = 2^{u+q-1} - 2^{u-1}$$

Next, m different functions $h_1, \ldots, h_m$ on $n-u-q$ variables will be used. It is needed to choose these functions in such a manner that for any $\tau \in GF(2)^m$, $nl(\bigoplus_{j=1}^{m} \tau_j h_j)$ is high. Mostly, bent functions as in Propositions 5.1.9 and 5.1.10 will be used. Now, look at the construction $S = (f_1, \ldots, f_m)$ where $f_i = h_i \oplus g_i$. For any $\tau \in GF(2)^m$, $\bigoplus_{j=1}^{m} \tau_j f_j(x)$ can be written as

$$\bigoplus_{j=1}^{m} \tau_j h_j \oplus \bigoplus_{j=1}^{m} \tau_j g_j.$$

This can be done since the set of variables are distinct. The input variables of g_j are $x_1, \ldots, x_{u+q}$ and the input variables of h_j are $x_{u+q+1}, \ldots, x_n$. Consider the 5 cases seperately.

1. Here, $u \leq n \leq u+m$. By the Corollary 5.1.3, it is possible to construct $(n = u+q, m, t, 2^{n-1} - 2^{u-1})$ function S. (Since $u \leq n \leq u+m$, $u > m > q$)

2. Let $u+m \leq n < u+2m$. Here, take the first 2^{m-1} rows of A^*, i.e., $q = m-1$. The functions g_j are on $(u+q = u+m-1)$-variables. Each function needs to be repeated $\frac{2^n}{2^{u+m-1}}$ times. Use functions h_j on $(n-u-m+1)$-variables which are constant functions. $nl(g_j) = 2^{u+m-2} - 2^{u-1}$. Hence, by Proposition 5.1.5,

$$\begin{aligned} nl(f_j) &= 2^{n-u-m+1}(2^{u+m-2} - 2^{u-1}) \\ &= 2^{n-1} - 2^{n-m}. \end{aligned}$$

3. Let $u+2m \leq n < u+3m$. Choose q such that $n-u-q = 2m$. The g_j's are on $(u+q)$-variables. Take m bent functions h_j each of $2m$ variables. It is clear that $nl(g_j) = 2^{u+q-1} - 2^{u-1}$ and $nl(h_j) = 2^{2m-1} - 2^{m-1}$. Then, by using Corollary 5.1.6,

$$nl(S) = 2^{n-1} - 2^{u+m-1}.$$

4. For $n \geq u+3m-1$ and $\pi = n-u-m+1$ is even, take $q = m-1$ and a set of bent functions on $(n-u-m+1)$-variables. Since $n-u-m+1 \geq 2m$, one gets a set of m bent functions as in Proposition 5.1.10.

$$nl(g_j) = 2^{u+m-2} - 2^{u-1} \text{ and } \; nl(h_j) = 2^{(n-u-m+1)-1} - 2^{\frac{n-u-m+1}{2}-1}.$$

Thus,

$$\begin{aligned} nl(S) &= 2^{n-u-m+1}(2^{u+m-2} - 2^{u-1}) + 2^{u+m-1}(2^{n-u-m} - 2^{\frac{n-u-m+1}{2}-1}) \\ &\quad -(2^{u+m-1} - 2^{u-1})(2^{n-u-m} - 2^{\frac{n-u-m-1}{2}}) \\ &= 2^{n-1} - 2^{n-m} + 2^{-1} - 2^{\frac{n+m+u-2}{2}} - 2^{-1} + 2^{\frac{n+m+u-3}{2}} \\ &\quad -2^{n-m-1} + 2^{\frac{u+n-m-3}{2}} \\ &= 2^{n-1} - 2^{n-m} - 2^{\frac{n+m+u-2}{2}} + 2^{\frac{n+m+u-3}{2}} - 2^{n-m-1} + 2^{\frac{u+n-m-3}{2}} \\ &= 2^{n-1} - \tfrac{3}{2}2^{n-m} - 2^{\frac{n+m+u-2}{2}} + 2^{\frac{n+m+u-2}{2}-\frac{1}{2}} \\ &= 2^{n-1} - 2^{\frac{n+u-m-1}{2}} \end{aligned}$$

5. For $n \geq u+3m$ and $\pi = n-u-m+1$ is odd, use $q = m-1$ and a set of bent functions on $(n-u-m)$-variables, say $b_1, \ldots, b_m$ as in Proposition 5.1.10. Note that $n-u-m \geq 2m$. The construction of h_j is as $h_j = x_n \oplus b_j$. Thus,

$$nl(g_j) = 2^{u+m-1} - 2^{u-1}.$$

and

$$nl(h_j) = 2^{(n-u-m+1)-1} - 2^{\frac{(n-u-m+1)-1}{2}}.$$

Then

$$nl(S) = 2^{n-1} - 2^{\frac{n+u-m}{2}}.$$

□

In Theorem 5.2.1, the nonlinearity property of the constructed function S is observed. Now, consider the algebraic degree of S in Theorem 5.2.2 [Pasalic & Maitra, 2002].

Theorem 5.2.2. *In reference to Theorem* 5.2.1*, the algebraic degree of the function S is given by,*

$$\left\{ \begin{array}{ll} 2 \leq \deg S \leq n-u+1, \ u \leq n < u+m & (1) \\ 2 \leq \deg S \leq m, \ u+m \leq n < u+2m & (2) \\ \deg S = m, \ u+2m \leq n < u+3m & (3) \\ \deg S = \frac{n-u-m+1}{2}, \ n \geq u+3m-1, \ \pi \ even & (4) \\ \deg S = \frac{n-u-m}{2}, n \geq u+3m, \pi \ odd & (5) \end{array} \right\}$$

Proof: Consider any nonzero linear combination of $(f_1, \ldots, f_m)$ and denote any nonzero linear combination of h_j's as h and that of g_j's as g. It is clear that $degS = degf = max(deg(h), deg(g))$ since h and g are functions on distinct set of input variables.

1. Here, f can be seen as the concatenation of 2^q linear functions $(0 \leq q < m)$ of u-variables each. The exact calculation of the algebraic degree will depend in a complicated way on the choice of the codewords from C.

However, it is clear that the function is always nonlinear and hence the algebraic degree must be greater than 2. Also, the function f will have degree at most $q+1$. Here, $q = n-u$ which gives the result.

2. Here, take $q = m-1$(in Theorem 5.2.1). Now, f can be seen as the 2^{n-u-q} times repetition of function g, where g is the concatenation of 2^q linear functions ($0 \leq q < m$) of u variables each(The functions g_j's are constructed by using the columns of the matrix A^* where the entries are codewords of length u bit.). The exact calculation of the algebraic degree will depend in a complicated way on the choice of the codewords from C. However, it is clear that the function is always nonlinear and hence $deg(f) \geq 2$. Furthermore, the function g will have degree at most $q+1 = m-1+1 = m$.

3. In this case, $deg(f) = max(deg(h), deg(g))$. Now, $deg(h) = m$ since it is considered $2m$-variable bent funcions with property as described in one of the previous propositions. Moreover, $deg(g)$ is at most $q+1$. Now, $u+2m \leq n < u+3m$, which gives $q < m$. So $deg(f) = m$.

4. In this case, since h is a $(n-u-m+1)$-variable bent function, $deg(h) = \frac{n-u-m+1}{2}$ by Proposition 5.1.10 and $deg(g) \leq q+1 = m$.

   Here $n \geq u+3m-1$, i.e., $n-u-m+1 \geq 2m$ which gives $\frac{n-u-m+1}{2} \geq m$. Then $deg(f) = \frac{n-u-m+1}{2}$.

5. In this case, since h is a $(n-u-m)$-variable bent function, $deg(h) = \frac{n-u-m}{2}$ by Proposition 5.1.10 and $deg(g) \leq q+1 = m$.

   Here, $n \geq u+3m$, i.e., $n-u-m \geq 2m$ which gives $\frac{n-u-m}{2} \geq m$. Thus, $deg(f) = \frac{n-u-m}{2}$.

□

5.3 Further Improvements

This section summarizes further improvement [Pasalic & Maitra, 2002] of nonlinearity of items 2 and 3 in Theorem 5.2.1 [Pasalic & Maitra, 2002].

5.3.1 Improvement of Item 2

In the case of item 2, h_j's are taken as $(n-u-m+1)$-variable constant functions, thus without getting any nonlinearity for the h_j's and the linear combinations of them. The following example shows chosing nonlinear functions instead of constant functions provides S-box with higher nonlinearity [Pasalic & Maitra, 2002].

Example 5.3.1. Consider the construction of a $(9, 3, 1)$ function and start with a $[4, 3, 2]$ linear code.

$$n = 9, m = 3, t = 1, u = 4 \text{ and } \; q = m - 1 = 2.$$

$$u + m = 4 + 3 \leq n = 9 < u + 2m = 4 + 6 = 10.$$

Then, this is the case item 2 of Theorem 5.2.1. Hence the nonlinearity is $2^{9-1} - 2^{9-3} = 192$. Thus, one can construct a $(9, 3, 1, 192)$ function.

This is because the functions h_1,h_2 and h_3 are taken as constant functions on $(n - m - u + 1)$-variables. The functions g_1, g_2 and g_3 are on $u + m - 1 = 4 + 3 - 1 = 6$ variables and the nonlinearity of any linear combination of them is

$$2^{u+m-2} - 2^{u-1} = 2^5 - 2^3 = 24.$$

But, if the following functions are used instead of constant functions,

$$\begin{aligned} h_1(y_1, y_2, y_3) &= y_1y_2 \oplus y_3 \\ h_2(y_1, y_2, y_3) &= y_2y_3 \oplus y_1 \\ h_3(y_1, y_2, y_3) &= y_3y_1 \oplus y_2 \end{aligned}$$

(Any nonzero linear combination of these functions will provide nonlinearity 2.) then any linear combination of f_1, f_2, f_3 has the nonlinearity

$$nl(f) = 2^3 \times 24 + 2^6 \times 2 - 2 \times 24 \times 2 = 224.$$

This provides $(9, 3, 1, 224)$ function.

5.3.2 Improvement of Item 3

In the case of item 3 of Theorem 5.2.1, $u + 2m \leq n < u + 3m$. The value of q is selected such that $n - u - q = 2m$ without chosing $q = m - 1$. The following example shows chosing $q = m - 1$ provides S-box with higher nonlinearity [Pasalic & Maitra, 2002].

Example 5.3.2. Consider the construction of $(36, 8, 5)$ function using a $[17, 8, 6]$ linear code.

$$n = 36, m = 8, t = 5, u = 17, q = n - u - 2m = 3$$

Since $u + 2m = 17 + 2 \times 8 \leq n = 36 < u + 3m$. So item 3 is considered. Then

$$nl(S) = 2^{n-1} - 2^{u+m-1} = 2^{35} - 2^{24}.$$

Instead of choosing $q = n - u - 2m$, take $q = m - 1$. Then, g_j's are $u + q = u + m - 1 = 24$-variable functions and h_j's are $n - u - q = 12$-variable functions.

$$nl(g_j) = 2^{u+q-1} - 2^{n-1} = 2^{19} - 2^{16}.$$

To construct h_j's, use the construction method of K. Nyberg [Nyberg, 1993]. Consider the mapping $H(V) = V^{-1}$ where $V \in GF(2)^p$ and p is even. It is known that the nonlinearity of H is $2^{p-1} - 2^{\frac{p}{2}}$. Thus, it is clear that one can construct a function $H : GF(2)^p \rightarrow GF(2)^r$ for even p and $r \leq p$ with nonlinearity $2^{p-1} - 2^{\frac{p}{2}}$.

Here, in this construction, since $m = 8$, one needs 8 different 12-variable highly nonlinear functions h_j. And by the method of K. Nyberg [Nyberg, 1993], it is possible to construct m functions $h_1, \ldots, h_m$ on $(n-u-m+1)$-variables such that any nonzero linear combination of the functions h_j has the nonlinearity $2^{n-u-m} - 2^{\frac{n-u-m+1}{2}}$. Then $nl(h) = 2^{11} - 2^6$ and nonlinearity of f_j is

$$nl(f_j) = 2^{12}(2^{19} - 2^{16}) + 2^{24}(2^{11} - 2^6) - 2(2^{19} - 2^{16})(2^{11} - 2^6) = 2^{35} - 2^{23}.$$

So one gets a $(36, 8, 5)$ function with nonlinearity $2^{35} - 2^{23}$.

Now, if p is odd, consider a function $h : GF(2)^{p-1} \rightarrow GF(2)^r$ with $r \leq p-1$. Since $p-1$ is even, $nl(h) = 2^{p-2} - 2^{\frac{p-1}{2}}$ [Nyberg, 1993]. The r outputs of the function h can be denoted as $h_1, \ldots, h_r$. Now, take the function $H : GF(2)^p \rightarrow GF(2)^r$ with r output columns as $x_p \oplus h_1$, $x_p \oplus h_2, \ldots, x_p \oplus h_r$. Then $nl(S) = 2^{p-1} - 2^{\frac{p+1}{2}}$.

Now, the items 2 and 3 of Theorem 5.2.1 are updated and stated as Theorem 5.3.3 [Pasalic & Maitra, 2002].

Theorem 5.3.3. *Given a linear $[u, m, t+1]$ linear code, it is possible to construct an $(n, m, t, nl(S))$ function $S = (f_1, \ldots, f_m)$ where*

$$nl(S) = \left\{ \begin{array}{l} 2^{n-1} - 2^{n-m}, u + m \leq n < u + 2m - 1 \quad (i) \\ 2^{n-1} - 2^{\frac{n+u-m+1}{2}}, \pi = n - u - m + 1 \ even, u + 2m - 1 \leq n < u + 3m - 3 \quad (ii) \\ 2^{n-1} - 2^{\frac{n+u-m+2}{2}}, \pi = n - u - m + 1 \ odd, u + 2m \leq n < u + 3m - 3 \quad (iii) \\ 2^{n-1} - 2^{u+m-1}, u + 3m - 3 \leq n \leq u + 3m \quad (iv) \end{array} \right\}$$

Proof: This theorem is only the update version of the items 2 and 3 of Theorem 5.2.1. For the cases i and iv, keep the same result as in Theorem 5.2.1.

For the case ii,

$$\begin{array}{rcl} u + 2m - 1 & \leq & n \\ u + 2m - 1 + (-m) & \leq & n + (-m) \\ m + u - 1 \leq n - m \Rightarrow m & \leq & n - m - u + 1 \end{array}$$

In this case, use $q = m - 1$. Then g_j's are on $(u + m - 1)$-variables and h_j's are on $(n - u - m + 1)$-variables. Then using Proposition 5.3.1, the result is obtained.

For the case *iii*, the similar result is obtained. Note that the same strategy as in items *ii* and *iii* in item *iv* could be used. However, for the range of n in item *iv*, the nonlinearity $2^{n-1} - 2^{u+m-1}$ supersedes the nonlinearity achievable using the approach of items *ii* and *iii*.

□

Now, one can update item i of Theorem 5.3.2 as follows:

In item i, $u + m \leq n < u + 2m - 1$, i.e., $1 \leq n - u - m + 1 < m$. If one likes to use the strategy as in items *ii* and *iii* of Theorem 5.3.2, some function $S : GF(2)^p \rightarrow GF(2)^r$ for $r > p$ with same nonlinearity is need to be constructed. There is no general strategy to construct such a function. Also, it is clear that for the cases $n - u - m + 1 = 1, 2$ there is no possibility to get any nonlinearity. So item i of Theorem 5.3.2 can be updated as follows:

Proposition 5.3.4. *Given a linear* $[u, m, t + 1]$ *code, it is possible to construct an* $(n, m, t, nl(S))$ *function* $S = (f_1, \ldots, f_m)$ *where*

$$nl(S) = \left\{ \begin{array}{ll} 2^{n-1} - 2^{n-m}, & u + m \leq n < u + m + 2 \quad (I) \\ 2^{n-1} - 2^{n-m} + 2^u \nu(n - u - m + 1, m), & u + m + 2 \leq n < u + 2m - 1 \quad (II) \end{array} \right\}$$

where $\nu(p, r)$ *is the maximum possible nonlinearity of a* p*-input,* r*-output function with* $3 \leq p < r$.

Now, the results of Theorem 5.2.1, Theorem 5.3.2 and Proposition 5.3.3 are summarized in Theorem 5.3.4 [Pasalic & Maitra].

Theorem 5.3.5. *Given a linear* $[u, m, t + 1]$ *code, it is possible to construct an*

$(n, m, t, nl(S))$ *function* $S = (f_1, \ldots, f_m)$ *where*

$$nl(S) = \left\{ \begin{array}{l} 2^{n-1} - 2^{u-1},\ u \leq n < u + m \\ 2^{n-1} - 2^{n-m}, u + m \leq n < u + m + 2 \\ 2^{n-1} - 2^{n-m} + 2^u \nu(n - u - m + 1, m),\ u + m + 2 \leq n < u + 2m - 1 \\ 2^{n-1} - 2^{\frac{n+u-m+1}{2}}, u + 2m - 1 \leq n < u + 3m - 3, \pi\ even \\ 2^{n-1} - 2^{\frac{n+u-m+2}{2}}, u + 2m \leq n < u + 3m - 3, \pi\ odd \\ 2^{n-1} - 2^{u+m-1}, u + 3m - 3 \leq n < u + 3m \\ 2^{n-1} - 2^{\frac{n+u-m-1}{2}}, n \geq u + 3m - 1,\ \pi\ even \\ 2^{n-1} - 2^{\frac{n+u-m}{2}}, n \geq u + 3m, \pi\ even \end{array} \right\}$$

5.4 An 13×4 S-box Construction

Let's construct a function $S(x) : GF(2)^{13} \rightarrow GF(2)^3$. Take $u = 5, m = 3, t = 1$. We need a $[5, 3, 2]$ linear code. We can take the linear code C with basis c_0, c_1 and c_2 where $c_0 = 11000$, $c_1 = 10100$ and $c_2 = 00101$. Then,

$$C = 00000, 01100, 11101, 10001, 11000, 10100, 00101, 01001.$$

Let β be a primitive element in $GF(2^3)$. Look at the mapping

$$\phi(a_0 + a_1\beta + a_2\beta^2) = a_0c_0 + a_1c_1 + a_2c_2$$

Then,

$$\begin{aligned} \phi(1) &= c_0 \\ \phi(\beta) &= c_1 \\ \phi(\beta^2) &= c_2 \\ \phi(\beta^3) &= \phi(\beta + 1) = c_0 + c_1 \\ \phi(\beta^4) &= \phi(\beta^2 + \beta) = c_1 + c_2 \\ \phi(\beta^5) &= \phi(\beta^2 + \beta + 1) = c_0 + c_1 + c_2 \\ \phi(\beta^6) &= \phi(\beta^2 + 1) = c_0 + c_2 \\ \phi(\beta^7) &= \phi(1) = c_0 \end{aligned}$$

Then the matrix A^* is:

$$A^* = \begin{bmatrix} c_0 & c_1 & c_2 \\ c_1 & c_2 & c_0 + c_1 \\ c_2 & c_0 + c_1 & c_1 + c_2 \\ c_0 + c_1 & c_1 + c_2 & c_0 + c_1 + c_2 \\ c_1 + c_2 & c_0 + c_1 + c_2 & c_0 + c_2 \\ c_0 + c_1 + c_2 & c_0 + c_2 & c_0 \\ c_0 + c_2 & c_0 & c_1 \end{bmatrix}$$

A^* is a 7×3 matrix where each codeword exists only once in any nonzero linear combination of columns. Let's take the first 2^q row of the matrix A^* where $0 \leq q \leq m-1$, i.e., $0 \leq q \leq 2$.

Take $q = 2$ and form the matrix D.

$$D = \begin{bmatrix} c_0 & c_1 & c_2 \\ c_1 & c_2 & c_0 + c_1 \\ c_2 & c_0 + c_1 & c_1 + c_2 \\ c_0 + c_1 & c_1 + c_2 & c_0 + c_1 + c_2 \end{bmatrix}$$

For each column of D, we construct the functions g_{j+1} for $j = 0, 1, 2$ by the method

$$g_{j+1}(y, x) = \bigoplus_{\tau \in GF(2)^2} (y_1 \oplus \tau_1)(y_2 \oplus \tau_2)(d_{[\tau],j} \cdot x).$$

g_{j+1} is a $(u+q)$-variable function, i.e., g_1, g_2 and g_3 are 7-variable funtions and $y \in GF(2)^2$ and $x \in GF(2)^5$. Let $y = y_1 y_2$ and $x = x_1 x_2 x_3 x_4 x_5$. Now, let's construct g_1, g_2 and g_3.

$$
\begin{aligned}
g_1(y,x) &= \bigoplus_{\tau \in GF(2)^2} (y_1 \oplus \tau_1)(y_2 \oplus \tau_2)(d_{[\tau],0} \cdot x) \\
&= y_1y_2(x_1 \oplus x_2) \oplus y_1(y_2 \oplus 1)(x_1 \oplus x_3) \\
&\quad \oplus (y_1 \oplus 1)y_2(x_3 \oplus x_5) \oplus (y_1 \oplus 1)(y_2 \oplus 1)(x_2 \oplus x_3) \\
&= y_1y_2x_1 \oplus y_1y_2x_2 \oplus y_1y_2x_1 \\
&\quad \oplus y_1y_2x_3 \oplus y_1x_1 \oplus y_1x_3 \oplus y_1y_2x_3 \oplus y_1y_2x_5 \oplus y_2x_3 \oplus y_2x_5 \\
&\quad \oplus y_1y_2x_2 \oplus y_1y_2x_3 \oplus y_1x_2 \oplus y_1x_3 \oplus y_2x_2 \oplus y_2x_3 \oplus x_2 \oplus x_3 \\
&= y_1y_2x_3 \oplus y_1y_2x_5 \oplus y_1x_1 \oplus y_2x_5 \oplus y_1x_2 \oplus y_2x_2 \oplus x_2 \oplus x_3
\end{aligned}
$$

$nl(g_1) = 2^6 - 2^4 = 48$ and g_1 is 1-resilient.

$$
\begin{aligned}
g_2(y,x) &= \bigoplus_{\tau \in GF(2)^2} (y_1 \oplus \tau_1)(y_2 \oplus \tau_2)(d_{[\tau],1} \cdot x) \\
&= y_1y_2(x_1 \oplus x_3) \oplus y_1(y_2 \oplus 1)(x_3 \\
&\quad \oplus x_5) \oplus (y_1 \oplus 1)y_2(x_2 \oplus x_3) \oplus (y_1 \oplus 1)(y_2 \oplus 1)(x_1 \oplus x_5) \\
&= y_1y_2x_1 \oplus y_1y_2x_3 \oplus y_1y_2x_3 \oplus y_1y_2x_5 \\
&\quad \oplus y_1x_3 \oplus y_1x_5 \oplus y_1y_2x_2 \oplus y_1y_2x_3 \\
&\quad \oplus y_2x_2 \oplus y_2x_3 \oplus y_1y_2x_1 \oplus y_1y_2x_5 \\
&\quad \oplus y_1x_1 \oplus y_1x_5 \oplus y_2x_1 \oplus y_2x_5 \oplus x_1 \oplus x_5 \\
&= y_1y_2x_3 \oplus y_1y_2x_2 \oplus y_1x_3 \oplus y_2x_2 \\
&\quad \oplus y_2x_3 \oplus y_1x_1 \oplus y_2x_1 \oplus y_2x_5 \oplus x_1 \oplus x_5
\end{aligned}
$$

$nl(g_2) = 2^6 - 2^4 = 48$ and g_2 is 1-resilient.

$$\begin{aligned} g_3(y,x) &= \bigoplus_{\tau \in GF(2)^2} (y_1 \oplus \tau_1)(y_2 \oplus \tau_2)(d_{[\tau],2} \cdot x) \\ &= y_1y_2(x_3 \oplus x_5) \oplus y_1(y_2 \oplus 1)(x_2 \oplus x_3) \\ &\quad \oplus (y_1 \oplus 1)y_2(x_1 \oplus x_5) \oplus (y_1 \oplus 1)(y_2 \oplus 1)(x_2 \oplus x_5) \\ &= y_1y_2x_3 \oplus y_1y_2x_5 \oplus y_1y_2x_2 \oplus y_1y_2x_3 \oplus y_1x_2 \\ &\quad \oplus y_1x_3 \oplus y_1y_2x_1 \oplus y_1y_2x_5 \oplus y_2x_1 \oplus y_2x_5 \\ &\quad \oplus y_1y_2x_2y_1y_2x_5 \oplus y_1x_2 \oplus y_1x_5 \oplus y_2x_2 \oplus y_2x_5 \oplus x_2 \oplus x_5 \\ &= y_1y_2x_1 \oplus y_1y_2x_5 \oplus y_1x_3 \oplus y_2x_1 \oplus y_1x_5 \oplus y_2x_2 \oplus x_2 \oplus x_5 \end{aligned}$$

$nl(g_3) = 2^6 - 2^4 = 48$ and g_3 is 1-resilient. So we get three 7-variable, 1-resilient Boolean functions with nonlinearity 48.

We want to construct $S = (f_1, f_2, f_3)$ where $f_i = g_i \oplus h_i$ where each f_i is an n-variable Boolean function, g_i is an $(u+q)$-variable Boolean function and h_i is an $(n-u-q)$-variable Boolean function, $i = 1, 2, 3$.

We have constructed the functions g_i's. Now, to construct f_i's, we need to construct $m = 3$ different h_i's. To get 3 different h_i, $n - u - q \geq 2m$ must be satisfied. Since $n = 13$, $u = 5$, $q = 2$ and $m = 3$, $6 \geq 6$ and so we can construct 3 different 6 variable h_i's which are bent.

Construction of f_1

$$h_1(x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}) = x_8x_{11} \oplus x_9x_{12} \oplus x_{10}x_{13} \oplus b_1$$

where b_1 is any function on the variables x_{11}, x_{12}, x_{13}.

Let's take $b_1(x_{11}, x_{12}, x_{13}) = x_{11}x_{13} \oplus x_{12}$. Then,

$$h_1(x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}) = x_8x_{11} \oplus x_9x_{12} \oplus x_{10}x_{13} \oplus x_{11}x_{13} \oplus x_{12}$$

$$g_1(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = x_1x_2x_5 \oplus x_1x_2x_7 \oplus x_1x_3 \oplus x_2x_7 \oplus x_1x_4 \oplus x_2x_4 \oplus x_4 \oplus x_5$$

Then $f_1 = g_1 \oplus h_1$ where $f_1 : GF(2)^{13} \rightarrow GF(2)$

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_{13}) &= x_1x_2x_5 \oplus x_1x_2x_7 \oplus x_1x_3 \oplus x_2x_7 \oplus x_1x_4 \oplus x_2x_4 \oplus x_8x_{11} \\ &\quad \oplus x_9x_{12} \oplus x_{10}x_{13} \oplus x_{11}x_{13} \oplus x_4 \oplus x_5 \oplus x_{12} \end{aligned}$$

Construction of f_2

$$h_2(x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}) = x_8x_{11} \oplus x_9x_{12} \oplus x_{10}x_{13} \oplus b_2$$

where b_2 is any function on the variables x_{11}, x_{12}, x_{13}.

Let's take $b_2(x_{11}, x_{12}, x_{13}) = x_{11} \oplus x_{12}$. Then,

$$\begin{aligned} h_2(x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}) &= x_8x_{11} \oplus x_9x_{12} \oplus x_{10}x_{13} \oplus x_{11} \oplus x_{12} \\ g_2(x_1, x_2, x_3, x_4, x_5, x_6, x_7) &= x_1x_2x_5 \oplus x_1x_2x_4 \oplus x_1x_5 \oplus x_2x_4 \oplus x_2x_5 \\ &\quad \oplus x_1x_3 \oplus x_2x_3 \oplus x_2x_7 \oplus x_3 \oplus x_7 \end{aligned}$$

Then $f_2 = g_2 \oplus h_2$ where $f_2 : GF(2)^{13} \rightarrow GF(2)$

$$\begin{aligned} f_2(x_1, x_2, \ldots, x_{13}) &= x_1x_2x_5 \oplus x_1x_2x_4 \oplus x_1x_5 \oplus x_2x_4 \oplus x_2x_5 \oplus x_1x_3 \oplus x_2x_3 \\ &\quad \oplus x_2x_7 \oplus x_8x_{11} \oplus x_9x_{12} \oplus x_{10}x_{13} \oplus x_{11} \oplus x_{13} \oplus x_3 \oplus x_7 \end{aligned}$$

Construction of f_3

$$h_3(x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}) = x_8x_{11} \oplus x_9x_{12} \oplus x_{10}x_{13} \oplus b_3$$

where b_3 is any function on the variables x_{11}, x_{12}, x_{13}.

Let's take $b_3(x_{11}, x_{12}, x_{13}) = x_{11}x_{12} \oplus x_{12}$. Then,

$$h_3(x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}) = x_8x_{11} \oplus x_9x_{12} \oplus x_{10}x_{13} \oplus x_{11}x_{12} \oplus x_{12}$$

$$g_3(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = x_1x_2x_3 \oplus x_1x_2x_7 \oplus x_1x_5 \oplus x_2x_3 \oplus x_1x_7 \oplus x_2x_4 \oplus x_4 \oplus x_7$$

Then $f_3 = g_3 \oplus h_3$ where $f_3 : GF(2)^{13} \rightarrow GF(2)$

$$\begin{aligned} f_3(x_1, x_2, \ldots, x_{13}) &= x_1x_2x_3 \oplus x_1x_2x_7 \oplus x_1x_5 \oplus x_2x_3 \oplus x_1x_7 \oplus x_2x_4 \oplus x_8x_{11} \\ &\quad \oplus x_9x_{12} \oplus x_{10}x_{13} \oplus x_{11}x_{12} \oplus x_4 \oplus x_7 \oplus x_{12} \end{aligned}$$

Then, $S(x_1, \ldots, x_{13}) = (f_1, f_2, f_3)$ is constructed. $nl(S) = 2^{12} - 2^7$ since $u + 2m \leq n < u + 3m$.

Note that, the bent function construction is the construction of S. Maitra and P. Sarkar [Maitra & Sarkar, 1999].

Chapter 6

Computational Results

In this chapter, we introduce our results for the construction method of Johansson and Pasalic [Johansson & Pasalic, 2000]. In Section 6.1, we first recall the restrictions on the choice of parameters. Then, we introduce the flowcharts of the programs that we use to construct t-resilient $n \times m$ S-boxes with nonlinearity $nl(S) = 2^{n-1} - 2^{n-d-1}$.

In Section 6.2 , we present our results for the nonlinearity of 1-resilient, 2-resilient and 3-resilient $n \times m$ S-boxes, which are constructed by our program for $n = 9$, $n = 10$ and $n = 11$. Then, we compare them with the results of [Johansson & Pasalic, 2000] and [Pasalic & Maitra, 2002].

To give an idea about the computational load of an exhaustive search algorithm, In Section 6.3, we present the tables which show the number N of $(n - d_{\max}, m)$ linear block codes and the number of all possible constructions.

For some cases, we have found S-boxes with higher nonlinearity than the highest nonlinearity achieved for the S-boxes in [Johansson & Pasalic, 2000]. For 2-resilient 9×2 S-box, the nonlinearity is found as 240 in [Johansson & Pasalic, 2000]. But we have shown that this is not possible and the highest nonlinearity that this construction can achieve is 224.

6.1 About Our Program

The construction in [Johansson & Pasalic, 2000] mainly depends on finding sufficient number of nonintersecting linear codes. For the construction of an $n \times m$ S-box with resiliency t, another parameter d should be chosen and the related number of required linear block codes is equal to $\left\lceil \frac{2^d}{2^m - 1} \right\rceil$. As mentioned in Chapter 4, there are some restrictions in the choice of parameters. To recall:

1. **Choice of "n" and "t"** The parameters n and t are restricted as: $n \geq 4$, $1 \leq t \leq n-3$.

2. **Choice of "d"** The parameter "d" must be chosen such that both $1 \leq d \leq n-t$ and $d \leq n-m$ are satisfied, i.e., $1 \leq d \leq \min\{n-m, n-t\}$. Since the nonlinearity of the S-box achieved by this construction method is $nl(S) = 2^{n-1} - 2^{n-d-1}$, to get high nonlinearity, d must be maximized. By this construction method, $n \times n$ S-boxes can not be constructed. Because, if $m = n$, then $d = 0$ and $nl(S) = 0$. According to the construction method, "d" must satisfy the inequality $\binom{n-d}{t+1} + \binom{n-d}{t+2} + \ldots + \binom{n-d}{n-d} \geq 2^d$.

3. **Number of Nonintersecting Linear Codes** For this construction method the cardinality of the set of $[n-d, m, t+1]$ nonintersecting linear codes must be $\left\lceil \frac{2^d}{2^m - 1} \right\rceil$. If the desired number of nonintersecting linear codes can not be found, then d is decreased by 1 and then searched again. But this time, nonlinearity falls.

4. **Singleton Bound** According to the Singleton Bound for a $[n-d, m, t+1]$ linear code, $t+1 \leq n-d-m+1$, i.e., $t \leq n-d-m$ must be satisfied. To get the results of this construction method, first we write a main program using Matlab programming tool which outputs the maximum possible value of d for the given values of n, m and t. Figure 6.1 shows the flowchart of this program. After finding the maximum value of d, the main program first tries to construct an $n \times m$ S-box with resilieny t accordingly. If the associated S-box does not exist, we decrease d by

1 and try again. The main program also gives the nonlinearities of all possible linear combinations of the constructed S-box with parameters n, m, t and d. Figure 6.2 shows the flowchart of the main program.

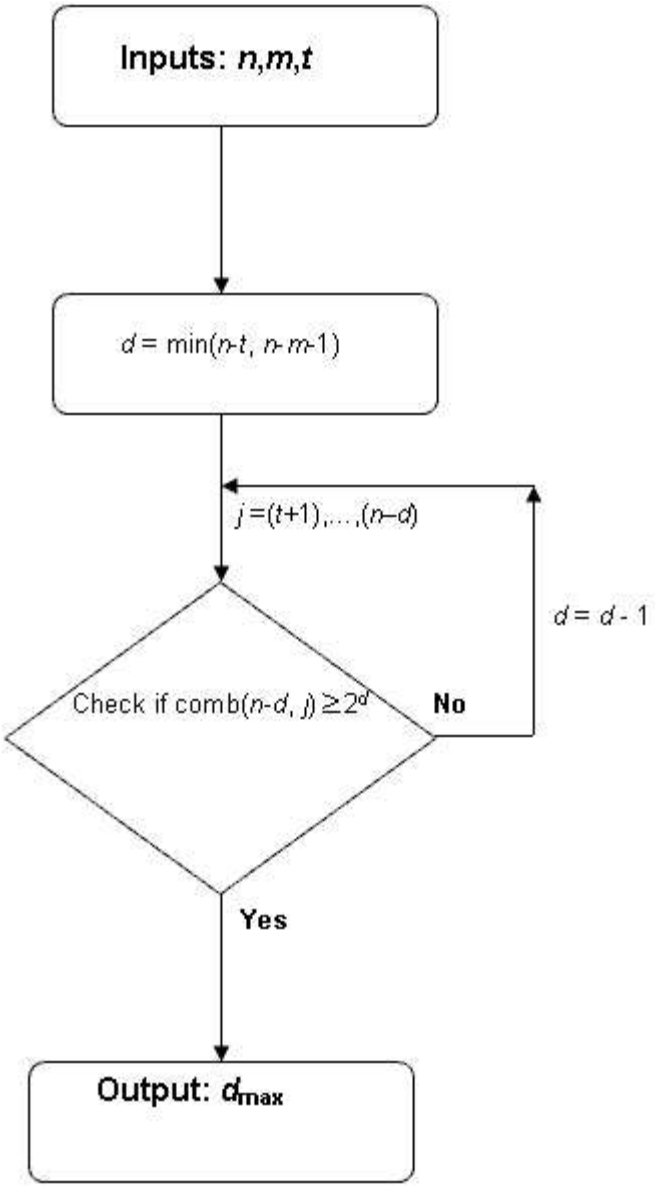

Figure 6.1: Flowchart of the Program Finding the Maximum Possible Value of d

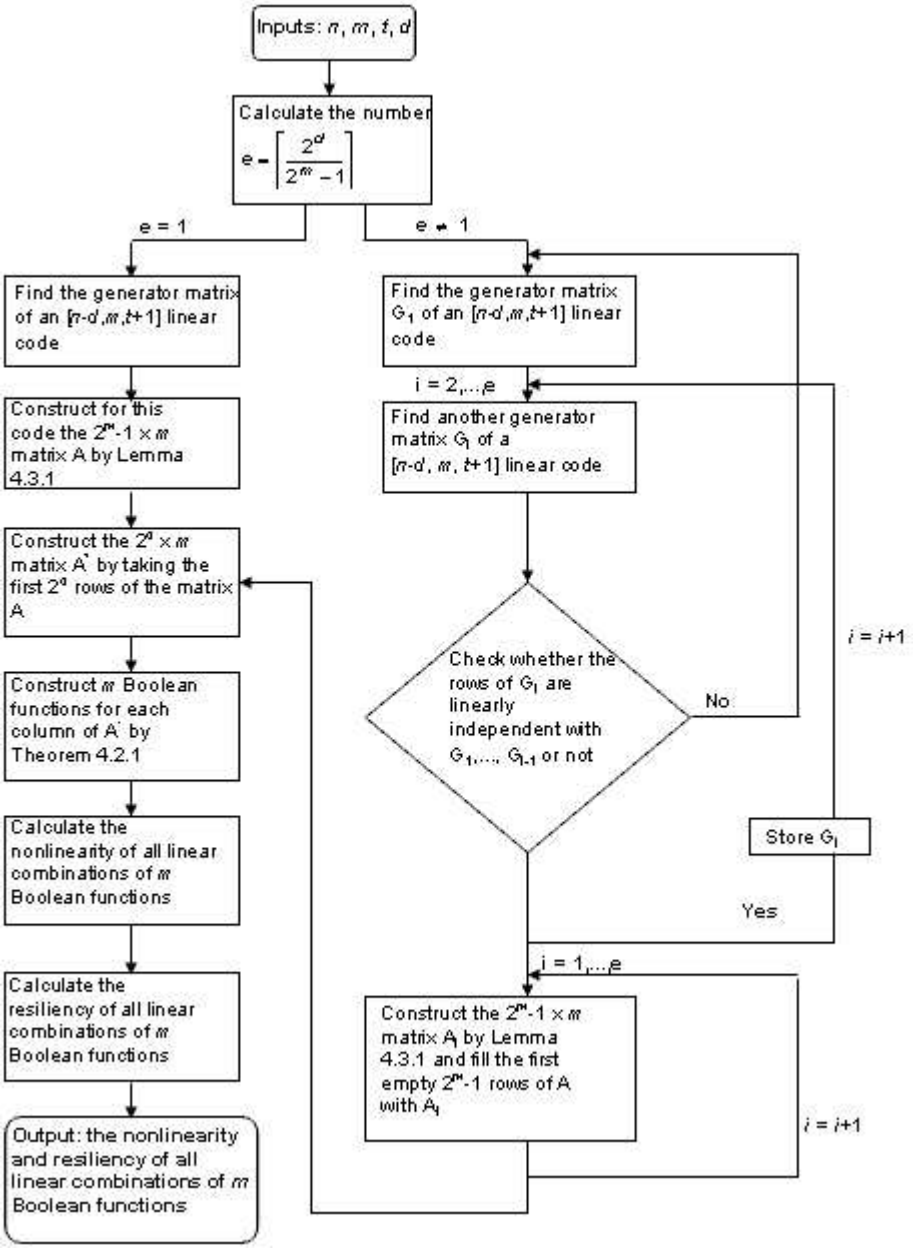

Figure 6.2: Flowchart of the Main Program

6.2 Our Results and Comparison

The value of the parameter d is important since it must be maximized in order to get high nonlinearity. Moreover, the number of the needed nonintersecting linear codes depends on the values of d and m. We give the maximum possible value $d_{\max}$, in addition the used values of d for the associated values of n, m and t.

Table 6.1 compares our highest achieved nonlinearity ($nl(S)$) results with the results in [Johansson & Pasalic, 2000] and in [Pasalic & Maitra, 2002] for 1-resilient $n \times m$ S-boxes using maximum possible value of $d_{\max}$. Note that $d_{\text{used}} < d_{\max}$ is the actual value of the parameter d, which is used in the construction whenever $d_{\max}$ does not yield a proper S-box. We emphasize the values of the nonlinearities which are higher than the other constructions. We refer to the previous construction in [Johansson & Pasalic, 2000] by "1^{st} Con." and the later construction in [Pasalic & Maitra, 2002] by "2^{nd} Con.". We have found the nonlinearity of the 1-resilient 10×5 S -box as 480, whereas the result in [Johansson & Pasalic, 2000] is 448.

Table 6.1: Highest Achieved Nonlinearity and d Values ($nl(S)/d_{\text{used}}/d_{\max}$) for 1-resilient $n \times m$ S-boxes

	$n = 9$			$n = 10$			$n = 11$		
m	Ours	1^{st} Con	2^{nd} Con	Ours	1^{st} Con	2^{nd} Con	Ours	1^{st} Con	2^{nd} Con
2	224/3/4	**240**	224	448/3/4	**480**	**480**	896/3/5	**992**	960
3	224/3/4	224	224	448/3/4	**480**	**480**	896/3/5	**992**	960
4	224/3/4	224	224	448/3/4	448	**480**	960/4/5	960	960
5	224/3/3	224	224	**480**/4/4	448	**480**	960/4/5	960	960
6	192/2/2 (res.1,3)	192	192	448/3/3	448	448	960/4/5	960	960

Table 6.2 compares our highest achieved nonlinearity ($nl(S)$) results with the results in [Johansson & Pasalic, 2000] and in [Pasalic & Maitra, 2002] for 2-resilient $n \times m$ S-boxes. We have found some results better than the others, such as the nonlinearity of the 2-resilient 9×3 S-box as 224, whereas the results in [Johansson & Pasalic, 2000] and in [Pasalic & Maitra, 2002] is 192. We have

found the nonlinearity of the 2-resilient 9×4 S-box as 192, that of 2-resilient 10×2 S-box as 480, that of the 2-resilient 10×4 S-box as 448, and that of the 2-resilient 11×3 S-box as 960, which are all better than the results given in either [Johansson & Pasalic, 2000] or [Pasalic & Maitra, 2002] or both, as can be observed from Table 6.2.

Table 6.2: Highest Achieved Nonlinearity and d Values ($nl(S)/d_{\text{used}}/d_{\max}$) for 2-resilient $n \times m$ S-boxes

	$n=9$			$n=10$			$n=11$		
m	Ours	1^{st} Con	2^{nd} Con	Ours	1^{st} Con	2^{nd} Con	Ours	1^{st} Con	2^{nd} Con
2	224/3/4	**240**	192	**480**/3/4	**480**	448	960/4/5	**992**	896
3	**224**/3/4	192	192	448/2/4	448	448	**960**/4/5	**960**	896
4	**192**/2/4	128	**192**	**448**/2/4	384	**448**	896/3/5	896	896
5	–	–	–	256/1/3	256	256	768/2/4	768	768
6	–	–	–	–	–	–	512/1/3	512	512

In the construction of 9×2 S-box, the nonlinearity found in [Johansson & Pasalic, 2000] is 240. But, we think that, this value can not be achieved by this method. According to the nonlinearity formula, nonlinearity is equal to 240 if and only if $d = 4$. But if $d = 4$, then there must exist 6 nonintersecting linear codes with parameters $[n-d, m, t+1] = [5, 2, 3]$. By computer search, we have found 5 nonintersecting linear codes. Moreover, we can see the reason by counting the codewords of 6 nonintersecting linear $[5, 2, 3]$ codes. In a $[5, 2, 3]$ linear code, there are 3 nonzero codewords. Therefore, for the construction which uses 6 nonintersecting $[5, 2, 3]$ linear codes, we need $3 \times 6 = 18$ nonzero codewords. Also, there are $2^5 = 32$ different words of length 5. Among them 1 is all zero word , 5 are of weight 1 and $\binom{5}{2} = 10$ are of weight 2. Then the number of words with weight at least 3 is $32 - (1 + 5 + 10) = 16$. So, there does not exist 6 nonintersecting $[5, 2, 3]$ linear codes. Then d can not be taken as 4. By decreasing d by 1, its maximum value is 3, hence the nonlinearity is 224.

Table 6.3 compares our highest achieved nonlinearity ($nl(S)$) results with the results in [Johansson & Pasalic, 2000] and in [Pasalic & Maitra, 2002] for 3-resilient $n \times m$ S-boxes. We have found the nonlinearity of the 3-resilient 9×2 S-box as 224, whereas the results in [Johansson & Pasalic, 2000] and in [Pasalic

& Maitra, 2002] is 192. We have also found the nonlinearity of the 3-resilient 10×2 S-box as 448, that of the 3-resilient 10×3 S-box as 448, that of the 3-resilient 10×4 S-box as 384, that of the 3-resilient 11×2 S-box as 960, that of the 3-resilient 11×4 S-box as 896 and that of the 3-resilient 11×6 S-box as 512, which are all better than [Johansson & Pasalic, 2000] and [Pasalic & Maitra, 2002].

Table 6.3: Highest Achieved Nonlinearity and d Values ($nl(S)/d_{\text{used}}/d_{\max}$) for 3-resilient $n \times m$ S-boxes

	$n = 9$			$n = 10$			$n = 11$		
m	Ours	1^{st} Con	2^{nd} Con	Ours	1^{st} Con	2^{nd} Con	Ours	1^{st} Con	2^{nd} Con
2	**224**/3/3	192	192	**448**/3/4	**448**	384	**960**/4/4	**960**	896
3	192/2/3	192	192	384/2/4	384	384	896/3/4	896	896
4	128/1/2	128	128	**384**/2/4	256	**384**	**896**/3/4	768	**896**
5	–	–	–	–	–	–	512/1/3	512	512
6	–	–	–	–	–	–	**512**/1/2	**512**	–

6.3 Number of Linear Block Codes in the Searched Space

The construction requires $e = \lceil \frac{2^d}{2^m - 1} \rceil$ nonintersecting linear block codes of dimension m, in the vector space $GF(2)^{n-d_{\max}}$. An exhaustive search would try all possible choices $(\binom{N}{e})$, where N is the number of subspaces of $GF(2)^{n-d_{\max}}$ of dimension m, which is shown by $N = N(n - d_{\max}, m)$ in equation (4.4) as the total number of $(n - d_{\max}, m)$ linear block codes. The parameter $(\binom{N}{e})$ is critical in determining the computational load of such an exhaustive search, and it is tabulated in Tables 6.4, 6.5 and 6.6, together with the d_{used} values of ours and Johansson and Pasalic's.

In fact, Tables 6.4, 6.5 and 6.6 show that; in the cases where the results of Johansson and Pasalic [Johansson & Pasalic, 2000] are better than ours, the search set is very large. There are also some cases that we get the same results with them in very large search sets. On the other hand, in almost all

the cases that our results are superior to theirs, the cardinality of the set in which nonintersecting block codes are searched, is very small. It seems very unlikely for a computer search algorithm to be unsuccessful in such sets of low cardinality. Hence, we conclude that the search method in their paper, should be a theoretical assignment in the set of some well-known linear block codes.

This theoretical choice seems to work quite well for the case of 1-resilient S-boxes shown in Table 6.4, where only one of our results(shown by bold letters) is superior to theirs, whereas 5 of their results(bold) are better than ours. However, in Table 6.6 there are 4 cases(bold) that our S-boxes have higher nonlinearity, and they all correspond to small search spaces.

Table 6.4: Number of Codes and d Values ($d_{\text{used}}/d_{\max}$) for 1-resilient $n \times m$ S-boxes

n	m	$d_{\max}$	$e = \left\lceil \frac{2^d}{2^m - 1} \right\rceil$	$N(m, n - d_{\max})$	$\binom{N}{e}$	d_{used} (ours)	d_{used} (Joh.& Pas.)
9	2	4	6	155	$1.7463e+010$	3	**4**
9	3	4	3	155	608685	3	3
9	4	4	2	31	465	3	3
9	5	3	1	63	63	3	3
9	6	2	1	127	127	2	2
10	2	4	6	651	$1.0330e+014$	3	**4**
10	3	4	3	1395	451478265	3	**4**
10	4	4	2	651	211575	3	3
10	5	4	1	63	63	**4**	3
10	6	3	1	127	127	3	3
11	2	5	11	651	$2.0481e+023$	3	**5**
11	3	5	5	1395	$4.3709e+013$	3	**5**
11	4	5	3	651	45770725	4	4
11	5	5	2	63	1953	4	4
11	6	4	1	127	127	4	4

Table 6.5: Number of Codes and d Values ($d_{\text{used}}/d_{\max}$) for 2-resilient $n \times m$ S-boxes

n	m	$d_{\max}$	$e = \left\lceil \frac{2^d}{2^m - 1} \right\rceil$	$N(m, n - d_{\max})$	$\binom{N}{e}$	d_{used} (ours)	d_{used} (Joh.& Pas.)
9	2	4	6	155	$1.7463e+010$	3	**4**
9	3	4	3	155	608685	**3**	2
9	4	3	1	651	651	**2**	1
10	2	4	6	651	$1.0330e+014$	3	3
10	3	4	3	1395	451478265	2	2
10	4	4	2	651	211575	2	1
10	5	3	1	2667	2667	1	1
11	2	5	11	651	$2.0481e+023$	4	**5**
11	3	5	5	1395	$4.3709e+013$	4	4
11	4	5	3	651	45770725	3	3
11	5	4	1	2667	2667	2	2
11	6	3	1	10795	10795	1	1

Table 6.6: Number of Codes and d Values ($d_{\text{used}}/d_{\max}$) for 3-resilient $n \times m$ S-boxes

n	m	$d_{\max}$	$e = \left\lceil \frac{2^d}{2^m - 1} \right\rceil$	$N(m, n - d_{\max})$	$\binom{N}{e}$	d_{used} (ours)	d_{used} (Joh.& Pas.)
9	2	3	3	651	45770725	**3**	2
9	3	3	2	1395	972315	2	2
9	4	2	1	11811	11811	1	1
10	2	4	6	651	$1.0330e+014$	3	3
10	3	4	3	1395	451478265	**3**	2
10	4	3	1	11811	11811	**2**	1
11	2	4	6	2667	$4.9701e+017$	4	4
11	3	4	3	11811	$2.7454e+011$	3	3
11	4	4	2	11811	69743955	**3**	2
11	5	3	1	97155	97155	1	1
11	6	2	1	788035	788035	1	1

Chapter 7

Conclusion

In this thesis, we have studied four S-box construction methods. Two of them are $n \times n$ S-box constructions presented by K. Nyberg [Nyberg, 1993]. For odd values of n, the inverse of the power polynomial $S(x) = x^{2^k+1}$, where k does not divide n, has the differential uniformity of 2, the nonlinearity $2^{n-1} - 2^{\frac{n-1}{2}}$ and the degree of $\frac{n+1}{2}$. The second construction of Nyberg [Nyberg, 1993] is the inversion mapping $S(x) = x^{-1}$, which was first noticed in 1957 by L. Carlitz and S. Uchiyama [Carlitz & Uchiyama, 1957]. The inversion mapping is also used in the Advanced Enryption Standard(AES) algorithm Rijndael for $n = 8$, and it has the differential uniformity of 4 for even and 2 for odd values of n, the nonlinearity greater than $2^{n-1} - 2^{\frac{n}{2}}$ and the degree of $n-1$. We review the theorems and propositions used in these two $n \times n$ S-box constructions following Nyberg, and in Proposition 3.1.1 and Proposition 3.2.1, we provide the details of the proofs to make them clearer.

The other two constructions are for $n \times m$ S-boxes, where $m < n$. Both of these methods utilize linear block codes. The construction method of T. Johansson and E. Pasalic [Johansson & Pasalic, 2000] depends on finding a set of nonintersecting linear codes and a full search in the set of linear block codes is the main problem of the method. The other construction [Pasalic & Maitra, 2002] is similar to the method in [Johansson & Pasalic, 2000], but it uses a single linear block code and many bent functions.

We have implemented the construction method of Johansson and Pasalic [Johansson & Pasalic, 2000] and we have found better results than those of both [Johansson & Pasalic, 2000] and [Pasalic & Maitra, 2002]. We have also shown that the highest possible nonlinearity achievable by Johansson and Pasalic construction for 2-resilient 9×2 S-box is 224; therefore, the nonlinearity value of 240 that is claimed to be found in [Johansson & Pasalic, 2000] is not possible. As can be observed from Table 6.1, the first construction [Johansson & Pasalic, 2000] seems to be more premising than the second construction [Pasalic & Maitra, 2002] in terms of the nonlinearity.

Comparing our construction results with those of [Johansson & Pasalic, 2000] as shown in Tables 6.4, 6.5 and 6.6, we notice that they have obtained better nonlinearities than ours for some cases, where the cardinality of the set of $(n-d, m, t+1)$ linear block codes is excessively large. There are also some cases that we get the same results with them in very large search sets, say of cardinality 10^{17}. On the other hand, it is quite interesting to observe that, in almost all the cases that our results are superior to theirs, the cardinality of the set in which nonintersecting block codes are searched, is very small. It seems very unlikely for a computer search algorithm to be unsuccessful in such sets of low cardinality. Hence, we conclude that the search method in their paper, should be a theoretical assignment in the set of some well-known linear block codes. Apparently, such an assignment may miss the possibilities, which can be caught by a full computer search algorithm.

References

[**Blahut**, 1983] Richard E. Blahut, "Theory and Practise of Error Control Codes"

[**Carlet**, 1990] C. Carlet, "Codes de Reed-Muller, Codes de Kerdock et de Preparata", Publication of LITP, Institut Blaise Pascal, Universiteé Paris.

[**Carlitz** & **Uchiyama**, 1957] L. Carlitz and S. Uchiyama, "Bounds for Exponential Sums", Duke Math. J. 24, pp. $37 - 41$, 1957.

[**Cheon**, 2001] J. H. Cheon, "Nonlinear Vector Resilient Functions", in Advances in Cryptology, Crypto 2001, Springer Verlag, 2001.

[**Friedman**, 1982] J. Friedman, "On the Bit Extraction Problem", Proc. $33rd$ IEEE Symp. Foundations of Computer Science, 1982, pp. 314-319.

[**Heys**, 2001] H. M. Heys, "A Tutorial on Linear and Differential Cryptanalysis", Technical Report CORR 2001-17, Centre for Applied Cryptographic Research, Department of Combinatorics and Optimization, University of Waterloo, March 2001.

[**Johansson** & **Pasalic**, 2000] T. Johansson and E. Pasalic, "A Construction of Resilient Functions with High Nonlinearity", Proc. IEEE Int. Symp. Information Theory, Sorrento, Italy, 2000.

[**Knudsen & Robshaw**, 1994] L. Knudsen and M.J.B. Robshaw, "Nonlinear Approximations in Linear Cryptanalysis", Advances in Cryptology, Eurocrypt 96, Springer Verlag, 1994, pp 1-11.

[**Kurosawa** & **Satoh** & **Yamamoto**, 1997] K. Kurosawa, T. Satoh and K. Yamamoto, "Highly Nonlinear $t-$resilient Functions", J. Univ. Comput. Sci., vol. 3, no. 6, pp. 721-729, 1997.

[**Lidl** & **Niederreiter**, 1986] R. Lidl, and H. Niederreiter, "Introduction to Finite Fields and Their Applications", 1986

[**Maitra & Sarkar**, 1999] S. Maitra, and P. Sarkar, "Highly Nonlinear Resilient Functions Optimizing Siegenthaler's Inequality", Advances in Cryptology, Proc. CRYPTO 99, Springer Verlag, 1999, pp. $198 - 215$.

[**Meier & Staffelbach**, 1989] W. Meier and O. Staffelbach, "Nonlinearity Criteria for Cryptographic Functions", Advances in Cryptology, Proc. Eurocrypt 89, Springer Verlag, 1989, pp. $549 - 562$.

[**Menezes, Oorschot & Vanstone,** 1997] A.Menezes, P.van Oorschot and S. Vanstone, "Handbook of Applied Cryptography", CRC Press,1997.

[**Nyberg**, 1990] K. Nyberg, "Constructions of Bent Functions and Difference Sets", Advances in Cryptology, Eurocrypt 90, Springer Verlag, 1991, vol. 473, pp. 151-160.

[**Nyberg**, 1992] K. Nyberg, "On the Construction of Highly Nonlinear Permutations", Advances in Cryptology, Eurocrypt 92, Springer Verlag, 1992, vol. 658, pp. 92-98.

[**Nyberg**, 1993] K. Nyberg, "Differentially Uniform Mappings for Cryptography", Advances in Cryptology, Eurocrypt 93, Springer Verlag, 1994, vol. 765, pp. 55-64.

[**Pasalic** & **Maitra**, 2002] E. Pasalic and S. Maitra, "Linear Codes in Generalized Construction of Resilient Functions With Very High Nonlinearity", IEEE Transactions on Information Theory, vol. 48, no. 8, 2002.

[**Sarkar & Maitra,** 2000] Palash Sarkar and Subhamoy Maitra, "Construction of Nonlinear Boolean Functions with Important Cryptographic Prop-

erties", Advances in Cryptology, Eurocrypt 2000, Springer Verlag 2000, vol. 1807, pp. 485-506.

[**Siegenthaler,** 1984] T. Siegenthaler, "Correlation-immunity of Nonlinear Combining Functions for Cryptographic Applications", IEEE Transactions on Inform. Th., vol. IT-30(5), 1984, pp. 776 − 780.

[**Stinson**, 1993] D. R. Stinson, "Resilient Functions and Large Sets of Orthogonal Arrays", Congressus Numerantium, vol. 92, pp. 105-110, 1993.

[**Stinson & Massey**, 1995] D. R. Stinson and J. L. Massey, "An Infinite Class of Counterexamples to a Conjecture Concerning Nonlinear Resilient Functions", J. Cryptology, vol. 8, no.3, pp. 168-173, 1995.

[**Wan**, 1993] Z. Wan, "Geometry of Classical Groups Over Finite Fields", Studentlitteratur, Lund, 1993.

[**Webster** & **Tavares**, 1985] A. F. Webster and S. E. Tavares, "On the Design of $S-$boxes", Advances in Cryptology, Crypto 85,Springer Verlag, 1986, pp. 523-534.

[**Zhang** & **Zheng**, 1997] X. M. Zhang and Y. Zheng, "Cryptographically Resilient Functions", IEEE Trans. Inform. Theory, vol. 43, pp. 1740-1747, 1997.