Lecture Notes on Cryptographic Boolean Functions

Anne Canteaut

Inria, Paris, France
Anne.Canteaut@inria.fr
https://www.rocq.inria.fr/secret/Anne.Canteaut/

version: March 10, 2016

Contents

Chapter 1

Boolean functions

1.1 Boolean functions and their representations

Definition 1.1 (Boolean function)**.** *A Boolean function of n variables is a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2$. Its* value vector *is the binary vector v_f of length 2^n composed of all $f(x)$ when $x \in \mathbb{F}_2^n$.*

1.1.1 Truth table and Algebraic normal form

A Boolean function is usually defined by its *truth table*, which gives the images of all elements in $\mathbb{F}_2^n$. For instance, Table 1.1 is the truth table of a Boolean function of 3 variables. The

x_1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1
$f(x_1, x_2, x_3)$	0	1	0	0	0	1	1	1

Table 1.1: Truth table of a Boolean function of 3 variables.

value vector of f is the vector of $\mathbb{F}_2^8$ corresponding to the last row in the truth table.

Boolean functions are often identified with their value vectors. In particular, the weight and the support of a Boolean function f refer to the weight and the support of its value vector v_f. Most cryptographic applications use *balanced* Boolean functions, i.e., Boolean functions f whose output is uniformly distributed. This equivalently means that the weight of v_f is half of its length.

Besides the truth table, there are several other representations of Boolean functions which may be more appropriate in some contexts. In coding theory and in cryptography, a very natural representation is the so-called *algebraic normal form* (ANF), which corresponds to the expression of a Boolean function as a multivariate polynomial. Since the n inputs of the function take their values in $\mathbb{F}_2$, they must be considered modulo $X^2 + X$. Therefore, this polynomial has degree at most 1 in each input variable. It follows that any monomial of this polynomial is the product of some input variables. Each monomial can then be characterized by a subset of $I = \{1, \ldots, n\}$, i.e., $\prod_{i \in I} x_i$, or equivalently by the n-bit vector u having I as support. This second notation will be extensively used in the context of Boolean functions.

Notation 1.2. For any $u \in \mathbb{F}_2^n$, x^u denotes the monomial in $\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2+x_1, \ldots, x_n^2+x_n)$ defined by

$$\prod_{i=1}^{n} x_i^{u_i} \ .$$

The following theorem then shows that any Boolean function can be uniquely represented by a multivariate polynomial, and it also gives a simple formula for computing this polynomial from the value vector of the function.

Theorem 1.3 (Algebraic normal form)**.** *Let f be a Boolean function of n variables. Then, there exists a unique multivariate polynomial in $\mathbb{F}_2[x_1, \ldots, x_n]/(x_1^2+x_1, \ldots, x_n^2+x_n)$ such that*

$$f(x_1, \ldots, x_n) = \sum_{u \in \mathbb{F}_2^n} a_u x^u, \ \textit{with } a_u \in \mathbb{F}_2 \ .$$

This multivariate polynomial is called the algebraic normal form (ANF) *of f.*

Moreover, the coefficients of the ANF and the values of f satisfy:

$$a_u = \sum_{x \preceq u} f(x) \ \textit{and} \ f(u) = \sum_{x \preceq u} a_x,$$

where the sums are in $\mathbb{F}_2$ and $x \preceq y$ if and only if $x_i \leq y_i$ for all $1 \leq i \leq n$.

Proof. We first show by induction on n that the ANF of an n-variable Boolean function can be uniquely computed from its truth table.

- For $n = 1$, it is easy to check that the polynomial $a_1 x + a_0$ with $a_1 = f(0) + f(1)$ and $a_0 = f(0)$ is the unique polynomial equal to f.
- Induction step. Given an n-variable function f, we consider the two $(n-1)$-variable Boolean functions g and h defined by

  $$g(x_1, \ldots, x_{n-1}) = f(x_1, \ldots, x_{n-1}, 0) \text{ and } h(x_1, \ldots, x_{n-1}) = f(x_1, \ldots, x_{n-1}, 1) \ .$$

  Then, we have

  $$f(x_1, \ldots, x_n) = (1 + x_n) g(x_1, \ldots, x_{n-1}) + x_n h(x_1, \ldots, x_{n-1})$$

  or equivalently,

  $$f(x_1, \ldots, x_n) = g(x_1, \ldots, x_{n-1}) + x_n \left(g(x_1, \ldots, x_{n-1}) + h(x_1, \ldots, x_{n-1})\right) \ .$$

  We apply the induction hypothesis and denote by α_u (resp. β_u) for $u \in \mathbb{F}_2^{n-1}$ the coefficients of the ANF of g (resp. of h). We know that

  $$\alpha_u = \sum_{x \preceq u} g(x) = \sum_{x \preceq u} f(x, 0) \text{ and } \beta_u = \sum_{x \preceq u} h(x) = \sum_{x \preceq u} f(x, 1) \ .$$

  We then deduce that

  $$\begin{aligned} f(x_1, \ldots, x_n) &= g(x_1, \ldots, x_{n-1}) + x_n \left(g(x_1, \ldots, x_{n-1}) + h(x_1, \ldots, x_{n-1})\right) \\ &= \sum_{u \in \mathbb{F}_2^{n-1}} \alpha_u \prod_{i=1}^{n-1} x_i^{u_i} + \sum_{u \in \mathbb{F}_2^{n-1}} (\alpha_u + \beta_u) \left(\prod_{i=1}^{n-1} x_i^{u_i}\right) x_n \ . \end{aligned}$$

Therefore, the coefficients a_v, $v = (v_1, \ldots, v_n) \in \mathbb{F}_2^n$, of the ANF of f are given by

$$a_v = \begin{cases} \alpha_{v_1,\ldots,v_{n-1}} & \text{if } v_n = 0 \\ \alpha_{v_1,\ldots,v_{n-1}} + \beta_{v_1,\ldots,v_{n-1}} & \text{if } v_n = 1 \end{cases}$$

From the expressions of the coefficients α and β, we deduce that

$$a_v = \begin{cases} \sum_{u \preceq (v_1,\ldots,v_{n-1})} f(u,0) & \text{if } v_n = 0 \\ \sum_{u \preceq (v_1,\ldots,v_{n-1})} f(u,0) + \sum_{u \preceq (v_1,\ldots,v_{n-1})} f(u,1) & \text{if } v_n = 1 \end{cases}$$

implying that

$$a_v = \sum_{u \preceq v} f(u) \ .$$

Conversely, the values of f are uniquely determined by its ANF since the function over $\mathbb{F}_2^{2^n}$ which maps the value vector of f to the vector of coefficients of its ANF is an involution. Indeed, for any $y \in \mathbb{F}_2^n$, we have

$$\begin{aligned} \sum_{u \preceq y} a_u &= \sum_{u \preceq y} \sum_{x \preceq u} f(x) \\ &= \sum_{x \preceq y} f(x) |\{u \in \mathbb{F}_2^n : x \preceq u \preceq y\}| \\ &= \sum_{x \preceq y} 2^{wt(y)-wt(x)} f(x) \ . \end{aligned}$$

All terms in this sum are then zero modulo 2 unless $x = y$. Thus

$$\sum_{u \preceq y} a_u = f(y) \ ,$$

which means that the transformation we consider is an involution. ◇

Example 1.1. Computing the ANF of the function defined in Table 1.1. Using the previous theorem, we compute the coefficients of the ANF of this Boolean function:

$$\begin{aligned} a_{000} &= f(000) = 0 \\ a_{100} &= f(100) + f(000) = 1 \\ a_{010} &= f(010) + f(000) = 0 \\ a_{110} &= f(110) + f(010) + f(100) + f(000) = 1 \\ a_{001} &= f(001) + f(000) = 0 \\ a_{101} &= f(101) + f(001) + f(100) + f(000) = 0 \\ a_{011} &= f(011) + f(001) + f(010) + f(000) = 1 \\ a_{111} &= \sum_{x \in \mathbf{F}_2^3} f(x) = wt(f) \bmod 2 = 0 \ . \end{aligned}$$

Thus, the ANF of f is

$$x_1 + x_1x_2 + x_2x_3 \ .$$

The *degree* of f is then the degree of the largest monomial in the ANF of f, i.e.,

$$\deg f = \max_{u \in \mathbb{F}_2^n : a_u \neq 0} wt(u) \ .$$

For instance the function considered in the previous example has degree 2.

It is worth noticing that there exist several other representations of Boolean functions which may be more convenient than the ANF in some other contexts. For instance, the disjunctive normal form represents the function by some products between variables and negations of variables, which are added by an OR. A disjunctive normal form of the function defined in Table 1.1 is

$$x_1\overline{x_2x_3} \,\mathrm{OR}\, x_1\overline{x_2}x_3 \,\mathrm{OR}\, \overline{x_1}x_2x_3 \,\mathrm{OR}\, x_1x_2x_3 \ .$$

Such a representation may be more appropriate than the ANF when we want to determine the smallest circuit which implements the function in a context where only AND, NOT and OR gates are available, see [Weg87] for more details.

1.1.2 Computing the Algebraic Normal Form

The general form of the transformation which associates the coefficients of the ANF to the value vector is

$$\begin{array}{rccc} \mathcal{M}_n : & \mathbb{F}_2^{2^n} & \to & \mathbb{F}_2^{2^n} \\ & a = (a_u, u \in \mathbb{F}_2^n) & \mapsto & (b_u, u \in \mathbb{F}_2^n) \quad \text{with } b_u = \sum_{v \preceq u} a_v \ . \end{array}$$

This function is called *the binary Möbius transform*. Indeed, Möbius inversion is a method for inverting sums over a partially ordered sets. This general inversion formula appears in many contexts in combinatorics. For instance, it leads to the principle of inclusion-exclusion and to the expression of Euler ϕ-function [Rot64, Moe12]. In our context, we have proved in Theorem 1.3 that the transformation $\mathcal{M}_n$ is an involution, so any algorithm for computing the binary Möbius transform can be used both for computing the ANF from the value vector and for computing the value vector from the ANF.

The naive method for computing $\mathcal{M}_n(a)$ consists in evaluating the sum of the coordinates a_v of a over all positions $v \preceq u$ for the 2^n successive elements $u \in \mathbb{F}_2^n$. Since the sum defining b_u has $2^{wt(u)}$ terms, the overall complexity of this algorithm is

$$\sum_{i=0}^{n} \binom{n}{i} 2^i = 3^n \ .$$

But there exists a faster algorithm for computing the image of an element by $\mathcal{M}_n$ which has complexity $n2^{n-1}$ only. This algorithm exploits the fact that if we decompose any vector $a = (a_u, u \in \mathbb{F}_2^n)$ into two halves, namely $L(a) = (a_{u,0}, u \in \mathbb{F}_2^{n-1})$ and $R(a) = (a_{u,1}, u \in \mathbb{F}_2^{n-1})$, we get the following recursive formula:

$$L(\mathcal{M}_n(a)) = \mathcal{M}_{n-1}(L(a)) \text{ and } R(\mathcal{M}_n(a)) = \mathcal{M}_{n-1}(L(a)) + \mathcal{M}_{n-1}(R(a))$$

where the addition denotes the addition in $\mathbb{F}_2^{2^{n-1}}$. The corresponding algorithm then starts from $(a_u, u \in \mathbb{F}_2^n)$ where the values of u are written in lexicographic order. The k-th step, for $1 \leq k \leq n$, then computes the images by $\mathcal{M}_k$ of the 2^{n-k} vectors of 2^k consecutive bits composing a. The result at Step k is then obtained from the result at Step $(k-1)$ by

splitting the vector into blocks of 2^k consecutive bits, and for each block, the first half of the block remains unchanged while the second half is replaced by the sum of both halves. This iterative process is described by Algorithm 1. In this algorithm, the vectors $(a_u, u \in \mathbb{F}_2^n)$ are equivalently represented by 2^n-bit arrays $(a[i], 0 \leq i < 2^n)$, where n-bit integers are identified with n-bit vectors.

Algorithm 1 Evaluating the Möbius transform $\mathcal{M}_n$.

Input: $(a[i], 0 \leq i < 2^n)$
Output: $b = \mathcal{M}_n(a)$
for i from 0 to $2^n - 1$ **do**
  $b[i] \leftarrow a[i]$
end for
for k from 1 to n **do**
  for i from 0 to 2^{n-k} **do**
    // Compute the image of the i-th 2^k-bit block under $\mathcal{M}_k$
    for j from 0 to $2^{k-1} - 1$ **do**
      $b[2^k i + 2^{k-1} + j] \leftarrow b[2^k i + j] + b[2^k i + 2^{k-1} + j] \bmod 2$
    end for
  end for
end for
return b

Example 1.2. Computing the ANF of a 3-variable Boolean function. Let us denote by $f[0], \ldots, f[7]$ the array representing the 8-bit value vector of the 3-variable Boolean function f, namely $f[i] = f(i_0, i_1, i_2)$ where $i = \sum_{j=0}^{2} i_j 2^i$. Then, the operations performed during the three successive steps of Algorithm 1 are described in Table 1.2.

Example 1.3. Computing the ANF of a 5-variable Boolean function in C. If the value vector of the function is stored as 32-bit integer `x`, then the corresponding ANF is computed by the following program.

```
x ^= (x & 0x55555555) << 1;
x ^= (x & 0x33333333) << 2;
x ^= (x & 0x0f0f0f0f) << 4;
x ^= (x & 0x00ff00ff) << 8;
x ^= x << 16;
```

A more general program for any number of variables is given in [Jou09, Page 287] and can be downloaded from `http://www.joux.biz/algcrypt/PROGRAMS/Walsh_9-2.html`.

1.1.3 Reed-Muller codes

Reed-Muller codes are named after Reed [Ree54] and Muller [Mul54]: Muller described the codes while Reed proposed a majority-logic decoding algorithm for them. Reed-Muller codes can be defined over $\mathbb{F}_q$ but we here focus on the binary case. Binary Reed-Muller codes can be defined very easily in terms of Boolean functions.

Table 1.2: Computing the ANF of a 3-variable Boolean function f with Algorithm 1.

i	0	1	2	3	4	5	6	7
$f[i]$	$f[0]$	$f[1]$	$f[2]$	$f[3]$	$f[4]$	$f[5]$	$f[6]$	$f[7]$
Step 1	$f[0]$	$\boldsymbol{f[0]+f[1]}$	$f[2]$	$\boldsymbol{f[2]+f[3]}$	$f[4]$	$\boldsymbol{f[4]+f[5]}$	$f[6]$	$\boldsymbol{f[6]+f[7]}$
Step 2	$f[0]$	$f[0]+f[1]$	$\boldsymbol{f[0]+f[2]}$	$\boldsymbol{f[0]+f[1]}$ $\boldsymbol{+f[2]+f[3]}$	$f[4]$	$f[4]+f[5]$	$\boldsymbol{f[4]+f[6]}$	$\boldsymbol{f[4]+f[5]}$ $\boldsymbol{+f[6]+f[7]}$
Step 3	$f[0]$	$f[0]+f[1]$	$f[0]+f[2]$	$f[0]+f[1]$ $+f[2]+f[3]$	$\boldsymbol{f[0]+f[4]}$	$\boldsymbol{f[0]+f[1]}$ $\boldsymbol{f[4]+f[5]}$	$\boldsymbol{f[0]+f[2]}$ $\boldsymbol{+f[4]+f[6]}$	$\boldsymbol{f[0]+f[1]}$ $\boldsymbol{+f[2]+f[3]}$ $\boldsymbol{+f[4]+f[5]}$ $\boldsymbol{+f[6]+f[7]}$

Definition 1.4 (Reed-Muller codes)**.** *Let n be a positive integer and r an integer such $0 \leq r \leq n$. The* r-th order binary Reed-Muller code of length 2^n*, denoted by $\mathcal{R}(r,n)$, is the set of the value vectors of all Boolean functions of n variables with degree at most r:*

$$\mathcal{R}(r,n) = \{(f(x), x \in \mathbb{F}_2^n), f : \mathbb{F}_2^n \to \mathbb{F}_2 \text{ with } \deg f \leq r\} .$$

In particular $\mathcal{R}(0,n)$ is composed of the all-zero and the all-one 2^n-bit words. It is also known as the *repetition code* of length 2^n. On the other extreme, $\mathcal{R}(n,n)$ contains all 2^n-bit words.

Reed-Muller codes satisfy the following simple properties.

Proposition 1.5. *Let n be a positive integer and r an integer such $0 \leq r \leq n$.*

1. *$\mathcal{R}(r,n)$ is a linear code;*
2. *The value vectors of all monomials of degree at most r form a basis of $\mathcal{R}(r,n)$;*
3. *The dimension of $\mathcal{R}(r,n)$ is*

$$\dim \mathcal{R}(r,n) = \sum_{i=0}^{r} \binom{n}{i} ;$$

4. *$\mathcal{R}(r-1,n) \subset \mathcal{R}(r,n)$.*

Example 1.4. $\mathcal{R}(1,3)$ is a linear code of length 8 and dimension $1+3=4$. A generator matrix of $\mathcal{R}(1,3)$ consists of the value vectors of all monomials of degree 0 or 1 in x_1, x_2, x_3:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} .$$

Indeed, the rows of this matrix are equal to the value vectors of the functions $f(x_1,x_2,x_3)$ with respective ANF 1, x_1, x_2 and x_3, where the inputs (x_1,x_2,x_3) of the value vectors are ordered lexicographically.

1.2 Weight of a Boolean function

For studying the properties of a Boolean function with a given degree, the following decomposition provides a useful tool since it allows recursive constructions (and proofs by induction). For any Boolean function f of n variables and degree at most r, there exist two functions of $n-1$ variables, namely g and h, with $\deg g \leq r$ and $\deg h \leq r-1$ such that

$$f(x_1,\ldots,x_n) = g(x_1,\ldots,x_{n-1}) + x_n h(x_1,\ldots,x_{n-1}) .$$

The value vector v_f of f can then be decomposed into two halves corresponding to the inputs with $x_n = 0$ (resp. with $x_n = 1$). Then, we get

$$v_f = (v_g | v_g + v_h) ,$$

where v_g and v_h denote the value vectors of g and h and belong to $\mathcal{R}(r,n-1)$ and $\mathcal{R}(r-1,n-1)$ respectively.

This decomposition corresponds to a general construction, named the $(u|u+v)$ construction, described by Plotkin [Plo60], combining two codes of the same length to derive a new code which is twice longer.

1.2.1 Minimum weight of a Boolean function in $\mathcal{R}(r, n)$

The minimum weight of a function in $\mathcal{R}(r, m)$ can be deduced from the previous decomposition.

Theorem 1.6. *Let n be a positive integer and r an integer such $0 \leq r \leq n$. Then, the smallest weight for a nonzero function of n variables and degree at most r is 2^{n-r}. Moreover, this bound is tight by the monomials of degree r.*

Proof. Clearly, any monomial of n variables and degree r has weight 2^{n-r}. Then, we only have to prove that any function of degree at most r has weight $wt(f) \geq 2^{n-r}$, by induction on n.

- For $n = 1$ and $r = 0$, the only nonzero constant function is the all-one function, which has weight 2. Moreover, the value vectors of all nonzero functions correspond to all nonzero vectors in $\mathbb{F}_2^2$, implying that the minimum weight for a function in $\mathcal{R}(1, 1)$ is 1.
- Induction step. We now use that any f in $\mathcal{R}(r, n)$ can be decomposed as

  $$f(x_1, \ldots, x_n) = g(x_1, \ldots, x_{n-1}) + x_n h(x_1, \ldots, x_{n-1}) \ ,$$

  i.e.,

  $$v_f = (v_g | v_g + v_h) \ ,$$

  with $v_g \in \mathcal{R}(r, n-1)$ and $v_h \in \mathcal{R}(r-1, n-1)$. Then,

  $$wt(f) = wt(g) + wt(g + h) \ .$$

  If h is the zero function, then $wt(f) = 2wt(g) \geq 2 \times 2^{(n-1)-r} = 2^{n-r}$ by induction hypothesis.

  If $h \neq 0$, then, we use that, for any m-bit vectors x and y,

  $$\begin{aligned} wt(x + y) &= wt(x) + wt(y) - 2|\operatorname{Supp} x \cap \operatorname{Supp} y| \\ &\geq wt(x) + wt(y) - 2wt(x) \\ &\geq wt(y) - wt(x) \ . \end{aligned}$$

  Hence,

  $$wt(f) = wt(g) + wt(g + h) \geq wt(h) \geq 2^{(n-1)-(r-1)}$$

  where the last inequality is deduced from the induction hypothesis.

◇

1.2.2 Weights of affine functions

Proposition 1.7. *All n-variable Boolean functions of degree 1 have weight 2^{n-1}.*

Proof. A function of degree 1 can be written as $f(x) = a \cdot x + \varepsilon$ with $a \in \mathbb{F}_2^n \setminus \{0\}$ and $\varepsilon \in \mathbb{F}_2$. Then,

$$wt(f) = \{x \in \mathbb{F}_2^n : f(x) = 1\} = \{x \in \mathbb{F}_2^n : a \cdot x = 1 + \varepsilon\} = \begin{cases} \langle a \rangle^\perp & \text{if } \varepsilon = 1 \\ \mathbb{F}_2^n \setminus \langle a \rangle^\perp & \text{if } \varepsilon = 0 \end{cases}$$

The support of a function of degree 1 is then a hyperplane or the complement of a hyperplane, and has size 2^{n-1}. ◇

.

1.2.3 Weights of functions of degree $(n-1)$

Obviously, half of the Boolean functions of n variables have degree n. These functions are characterized by the following property.

Proposition 1.8. *A Boolean function f of n variables has an odd weight if and only if it has degree n.*

Proof. This comes directly from Theorem 1.3 which shows that the coefficient of degree n in the ANF of an n-variable Boolean function f is equal to the parity of the weight of its value vector. ◇

This implies that Boolean functions with maximal degree cannot be used in most cryptographic applications since their output distribution is biased.

1.2.4 Weights of functions of degree 2

The list of all possible weights for a Boolean function of degree 2 is also known [SB70]. The proof can be found in [SB70] or in Chapter 15,§ 2 of [MS77].

Proposition 1.9. *The weights of the Boolean functions of n variables with degree 2 are of the form*

$$w = 2^{n-1} \text{ or } w = 2^{n-1} \pm 2^{n-1-h} \text{ with } 0 \leq h \leq \left\lfloor \frac{n}{2} \right\rfloor .$$

1.2.5 Weight divisibility of Boolean functions

Another information is that the weight of any n-variable Boolean function of degree at most r is divisible by some power of 2 whose exponent depends on r and n.

Proposition 1.10. *Let n be a positive integer and r an integer such $0 < r \leq n$. Then, the weight of any n-variable Boolean function of degree at most r is divisible by*

$$2^{\lceil \frac{n}{r} \rceil - 1} .$$

This theorem was originally proved by Solomon and McEliece [SM66], but it is usually presented as a consequence of a more general theorem due to McEliece [McE72] on the divisibility of the weight of cyclic codes. A simpler proof can be derived from a classical formula for computing the weight of a Boolean function from its ANF (see e.g. [MM94] or [CHLL97, Page 240]).

1.3 Walsh transform

1.3.1 Definitions

Definition 1.11. *The* bias *(aka,* correlation*, or* imbalance*) of an n-variable Boolean function f is*

$$\mathcal{E}(f) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)} = 2^n - 2wt(f) .$$

In other words,

$$\Pr_X[f(X) = 1] = \frac{wt(f)}{2^n} = \frac{1}{2}\left(1 - \frac{\mathcal{E}(f)}{2^n}\right) .$$

Most notably, a Boolean function f is balanced if and only if $\mathcal{E}(f) = 0$.
In the following, we denote by φ_a the linear Boolean function $x \mapsto a \cdot x$.
We now reformulate Prop. 1.8 in terms of biases of linear functions.

Lemma 1.12. *Let $a \in \mathbb{F}_2^n$. Then*

$$\mathcal{E}(\varphi_a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{a \cdot x} = \begin{cases} 0 & \text{if } a \neq 0 \\ 2^n & \text{if } a = 0 \end{cases}.$$

A useful tool for studying Boolean functions is the Walsh transform.

Definition 1.13 (Walsh transform)**.** *Let f be a Boolean function of n variables. The Walsh transform of f is the function*

$$\begin{array}{ccc} \mathbb{F}_2^n & \to & \mathbb{Z} \\ a & \mapsto & \mathcal{E}(f + \varphi_a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) + a \cdot x} \end{array}.$$

The value $\mathcal{E}(f + \varphi_a)$ is called the Walsh coefficient *of f at point a and the multiset composed of all Walsh coefficients of f is called the* Walsh spectrum of f*.*

The Walsh spectrum of f corresponds to the biases of all approximations of f by a linear function. This quantity, especially its maximum, plays a major role in linear cryptanalysis.

1.3.2 Computing the Walsh transform.

A useful property of the Fourier transform is that it can be computed by a fast algorithm. While the naive method for computing all values of the Walsh transform of an n-variable Boolean function has complexity 2^{2n}, the complexity of the fast algorithm described in Algorithm 2 is proportional to $n2^n$.

Algorithm 2 Evaluating the Walsh transform of an n-variable Boolean function f.

Input: $(f[a], 0 \leq a < 2^n)$
Output: $(e[a] = \mathcal{E}(f + \varphi_a), 0 \leq a < 2^n)$
for i from 0 to $2^n - 1$ **do**
  $e[i] \leftarrow (-1)^{f[i]}$
end for
for k from 1 to n **do**
  for i from 0 to 2^{n-k} **do**
    // Compute the image of the i-th 2^k-bit block
    for j from 0 to $2^{k-1} - 1$ **do**
      $e'[2^k i + j] \leftarrow e[2^k i + j] + e[2^k i + 2^{k-1} + j] \bmod 2$
      $e'[2^k i + 2^{k-1} + j] \leftarrow e[2^k i + j] - e[2^k i + 2^{k-1} + j] \bmod 2$
    end for
  end for
  $e \leftarrow e'$
end for
return e

Example 1.5. Let us compute the Walsh transform of the 3-variable function

$$f(x_1, x_2, x_3) = x_1 + x_1x_2 + x_2x_3.$$

$f(a)$	0	1	0	0	0	1	1	1
$(-1)^{f(a)}$	1	-1	1	1	1	-1	-1	-1
step 1	0	2	2	0	0	2	-2	0
step 2	2	2	-2	2	-2	2	2	2
$\mathcal{E}(f+\varphi_a)$	0	4	0	4	4	0	-4	0

We deduce that the highest bias (in magnitude) of a linear approximation of this function is 4 and is obtained for instance for $\mathcal{E}(f + x_1)$. This equivalently means

$$\Pr[f(x_1, x_2, x_3) + x_2 = 0] = \frac{1}{2}\left(1 + \frac{4}{2^3}\right) = \frac{3}{4}\ .$$

1.3.3 Basic properties of the Walsh transform

Since it corresponds to a discrete Fourier transform, The Walsh transform enjoys all mathematical properties of a Fourier transform. For instance, the Walsh transform is (up to a constant factor) an involution.

Proposition 1.14. *Let f be a Boolean function of n variables. For all $b \in \mathbb{F}_2^n$, we have*

$$\sum_{a\in\mathbb{F}_2^n} (-1)^{a\cdot b}\mathcal{E}(f+\varphi_a) = 2^n(-1)^{f(b)}\ .$$

Proof.

$$\begin{aligned}\sum_{a\in\mathbb{F}_2^n} (-1)^{a\cdot b}\mathcal{E}(f+\varphi_a) &= \sum_{a\in\mathbb{F}_2^n}\sum_{x\in\mathbb{F}_2^n} (-1)^{a\cdot b}(-1)^{f(x)+a\cdot x} \\ &= \sum_{x\in\mathbb{F}_2^n} (-1)^{f(x)} \sum_{a\in\mathbb{F}_2^n} (-1)^{a\cdot(x+b)} \\ &= \sum_{x\in\mathbb{F}_2^n} (-1)^{f(x)}\mathcal{E}(\varphi_{x+b}) = 2^n(-1)^{f(b)}\end{aligned}$$

where the last equality is derived from Lemma 1.12 which states that $\mathcal{E}(\varphi_{x+b}) = 0$ unless $x = b$. ◇

As any Fourier transform, the Walsh transform satisfies the Parseval relation.

Proposition 1.15 (Parseval relation)**.** *Let f be a Boolean function of n variables. Then,*

$$\sum_{a\in\mathbb{F}_2^n} [\mathcal{E}(f+\varphi_a)]^2 = 2^{2n}\ .$$

Proof.

$$\begin{aligned}\sum_{a\in\mathbb{F}_2^n}\mathcal{E}^2(f+\varphi_a) &= \sum_{a\in\mathbb{F}_2^n}\left(\sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)+a\cdot x}\right)\left(\sum_{y\in\mathbb{F}_2^n}(-1)^{f(y)+a\cdot y}\right)\\ &= \sum_{x\in\mathbb{F}_2^n}\sum_{y\in\mathbb{F}_2^n}(-1)^{f(x)+f(y)}\left(\sum_{a\in\mathbb{F}_2^n}(-1)^{a\cdot(x+y)}\right)\\ &= 2^n\sum_{x\in\mathbb{F}_2^n}(-1)^{f(x)+f(x)} = 2^{2n}\end{aligned}$$

where the last-but-one equality is derived from Lemma 1.12. ◇

1.4 Linearity of a Boolean function

Since linear cryptanalysis exploits highly biased linear approximations, a natural question for a designer who needs to choose a good nonlinear function is to know the smallest possible value that can be achieved for the bias of the best linear approximation. This value is determined by the so-called *linearity* of the function, in the sense of the following definition.

Definition 1.16 (Linearity of a Boolean function)**.** *Let f be a Boolean function of n variables. The* linearity *of f is the highest magnitude of its Walsh coefficients, i.e.,*

$$\mathcal{L}(f) = \max_{a\in\mathbb{F}_2^n}|\mathcal{E}(f+\varphi_a)| \ .$$

1.4.1 Optimal linearity

The following lower bound on the linearity of a Boolean function is directly derived from Parseval relation.

Proposition 1.17. *Let f be a Boolean function of n variables. Then*

$$\mathcal{L}(f) \geq 2^{\frac{n}{2}} \ .$$

The functions for which equality holds are called bent functions*. They exist for even n only, and are not balanced.*

Proof. We first observe that a function f satisfying $\mathcal{L}(f) < 2^{\frac{n}{2}}$ cannot exist. Otherwise, from Parseval relation, we would have

$$2^{2n} = \sum_{a\in\mathbb{F}_2^n}\mathcal{E}^2(f+\varphi_a) < 2^n \times 2^n \ ,$$

a contradiction.

Obviously, this bound on the linearity is tight when n is even only. Let now f be a bent function. Parseval relation implies that, for all $a \in \mathbb{F}_2^n$,

$$\mathcal{E}^2(f+\varphi_a) = 2^n \ .$$

In other words, the Wash transform of a bent function has constant magnitude. In particular, for $a = 0$, we get that $\mathcal{E}(f) = \pm 2^{\frac{n}{2}}$ implying that f is not balanced. ◇

Since their output is biased, bent functions are of little use in cryptography. Most notably, any linear combination of the coordinates of a permutation is balanced, and then satisfies $\mathcal{L}(f) > 2^{\frac{n}{2}}$.

When n is odd, bent functions do not exist, and the lowest possible value for $\mathcal{L}(f)$ is unknown for $n \geq 9$. The quadratic function of $(2t+1)$ variables

$$f(x_1, \ldots, x_{2t+1}) = x_1x_2 + x_3x_4 + \ldots + x_{2t-1}x_{2t} + x_{2t+1}$$

satisfies

$$\mathcal{L}(F) = 2^{\frac{n+1}{2}} \ .$$

We then deduce the following proposition.

Proposition 1.18. *The lowest linearity for a Boolean function of n variables, n odd, satisfies*

$$2^{\frac{n}{2}} < \min_{f \in \mathsf{Bool}_n} \mathcal{L}(f) \leq 2^{\frac{n+1}{2}}$$

where the upper bound is tight for $n \leq 7$ and is not tight for $n \geq 9$.

The fact that the previous upper bound is not tight for $n \geq 9$ was a long-standing open problem solved in 2006 by [KMY07]. More precisely, Table 1.3 gives the lowest possible linearities for an n-variable Boolean function for n odd, $5 \leq n \leq 15$. Note that, by definition, the values of the Walsh transform, including the linearity, are always even.

n	5	7	9	11	13	15
$\min_{f \in \mathsf{Bool}_n} \mathcal{L}(f)$	8	16	$24-30$	$46-60$	$92-120$	$182-216$

Table 1.3: Smallest possible linearity for an n-variable Boolean function, where $a-b$ means that the lowest linearity can be any even integer in this range.

As previously mentioned, most cryptographic applications require the use of balanced functions. The lowest linearity for a balanced function of n variables is also unknown for n even. Only an upper bound on its value is derived from a recursive construction due to Dobbertin [Dob94]. Table 1.4 gives the lowest possible linearities for an n-variable *balanced* Boolean function for $4 \leq n \leq 10$. It is worth noticing that, since the weight of a balanced function is even, its degree is at most $(n-1)$. Then, all $(f + \varphi_a)$ have degree at most $(n-1)$ and an even degree, implying that their biases are divisible by 4.

n	4	5	6	7	8	9	10
$\min_{f \in \mathsf{Bool}_n} \mathcal{L}(f)$	8	8	12	16	$\{20, 24\}$	$\{24, 28, 32\}$	$\{36, 40\}$

Table 1.4: Smallest possible linearities for an n-variable balanced function.

1.4.2 Link with Reed-Muller codes.

The previously mentioned problems on the lowest possible linearity for a Boolean function have been extensively investigated in coding theory since they are related to the determination of the covering radius of the first-order Reed-Muller code. Indeed, the Walsh coefficients of a given Boolean function are determined by the weights of the coset of $\mathcal{R}(1,n)$ defined by the value vector of f, as detailed in the following proposition.

Proposition 1.19. *Let f be a Boolean function of n variables. Then, the Hamming weights of all functions in the set $f+\mathcal{R}(1,n)$ are*

$$\{2^{n-1}-\frac{1}{2}\mathcal{E}(f+\varphi_a); 2^{n-1}+\frac{1}{2}\mathcal{E}(f+\varphi_a),\ a\in\mathbb{F}_2^n\}\ .$$

In particular, the Hamming distance of f to the set of all affine functions, called the nonlinearity of f*, is given by*

$$d(f,\mathcal{R}(1,n)) = 2^{n-1}-\frac{1}{2}\mathcal{L}(f)\ .$$

Proof. By definition of $\mathcal{R}(1,n)$, the weights of $f+\mathcal{R}(1,n)$ correspond to the weights of all functions $c=f+\varphi_a+\varepsilon$, $a\in\mathbb{F}_2^n, \varepsilon\in\mathbb{F}_2$. If $\varepsilon=0$, we get that

$$wt(f+\varphi_a) = 2^{n-1}-\frac{1}{2}\mathcal{E}(f+\varphi_a)\ .$$

If $\varepsilon=1$, we have

$$wt(f+\varphi_a+1) = 2^n - wt(f+\varphi_a) = 2^{n-1}+\frac{1}{2}\mathcal{E}(f+\varphi_a)\ .$$

The expression of $d(f,\mathcal{R}(1,n))$ directly follows. ◇

Finding a function with minimal linearity then boils down to finding a value vector which lies as far as possible from the code $\mathcal{R}(1,n)$. This corresponds to the well-known notion of *covering radius*.

Definition 1.20 (Covering radius)**.** *Let $\mathcal{C}$ be a code of length n. The covering radius of $\mathcal{C}$ is the highest Hamming distance between $\mathcal{C}$ and a word in $\mathbb{F}_2^n$:*

$$\rho(\mathcal{C}) = \max_{c\in\mathbb{F}_2^n} d(c,\mathcal{C})\ .$$

Therefore, most of the previously mentioned results on the best linearity for a Boolean functions have been first proved in terms of covering radius of the first-order Reed-Muller codes, e.g. [Myk80, PW83, Hou93, Hou96b, Hou96a].

1.4.3 Affine equivalence for Boolean functions

The number of Boolean functions of n variables is 2^{2^n}, which makes very difficult to find functions with good cryptographic properties, like a high nonlinearity, for practical values of n. A useful tool for avoiding examining all functions is provided by any equivalence relation under which the involved properties are invariant. The most prominent equivalence relation is called *affine equivalence*.

Definition 1.21 (Affine equivalence)**.** *Two n-variable Boolean functions are said to be* affine equivalent *if there exists an affine permutation A of $\mathbb{F}_2^n$ such that*

$$g(x) = f(A(x)) \ .$$

Obviously, the degree is invariant under affine equivalence. This is also the case of the unsigned Walsh spectrum (and then of the linearity) as shown in the following proposition.

Proposition 1.22. *Let f and g be two n-variable Boolean functions such that $g(x) = f(Mx + a)$ for some invertible $n \times n$ matrix M and $a \in \mathbb{F}_2^n$. Then, for any $\alpha \in \mathbb{F}_2^n$,*

$$\mathcal{E}(g + \varphi_\alpha) = (-1)^{\alpha \cdot M^{-1} a} \mathcal{E}(f + \varphi_{(M^{-1})^T \alpha}) \ .$$

Proof.

$$\mathcal{E}(g + \varphi_\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{g(Mx+a)+\alpha \cdot x} = (-1)^{\alpha \cdot M^{-1} a} \sum_{y \in \mathbb{F}_2^n} (-1)^{g(y)+\alpha \cdot (M^{-1} y)} \ .$$

For any $y \in \mathbb{F}_2^n$, we have

$$\begin{aligned} \alpha \cdot (M^{-1} y) &= \sum_{i=1}^{n} \alpha_i (M^{-1} y)_i \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i M^{-1}_{ij} y_j \\ &= \sum_{j=1}^{n} y_j \left(\sum_{i=1}^{n} \alpha_i M^{-1}_{ij} \right) = y \cdot \left((M^{-1})^T \alpha \right) \ . \end{aligned}$$

We then deduce

$$\mathcal{E}(g + \varphi_\alpha) = (-1)^{\alpha \cdot M^{-1} a} \sum_{y \in \mathbb{F}_2^n} (-1)^{g(y) + y \cdot \left((M^{-1})^T \alpha \right)} = (-1)^{\alpha \cdot M^{-1} a} \mathcal{E}(g + \varphi_{(M^{-1})^T \alpha}) \ .$$

◇

Since they mainly focus on the linearity (or on the magnitude of the Walsh coefficients), all existing classifications of Boolean functions are up to affine equivalence. For instance, Berlekamp and Welch [BW72] have classified all functions up to 5 variables.

1.5 Existence of an approximation with fewer variables

Some divide-and-conquer attacks, like (fast) correlation attacks against combination keystream generators, exploit the fact that the involved Boolean function f can be approximated by a function g depending on fewer variables. Typically, in a combination generator composed of n LFSRs, if the combining function f can be approximated by a function g of $k < n$ variables, then the attack involves the initial states of only k out of the n constituent LFSRs.

1.5.1 Correlation-immunity order

A natural counter-measure to avoid these attacks consists then in using as a building-block a *correlation-immune function* in the sense of the following definition.

Definition 1.23 (Correlation-immunity [Sie84])**.** *A Boolean function f is* t-th order correlation-immune *if the probability distribution of its output is unaltered when any t input variables are fixed. Balanced t-th order correlation-immune functions are called* t-resilient*.*

Note that a t-th order correlation-immune function is k-th order correlation-immune for any $k \leq t$. The correlation-immunity order of a function f then implicitly refers to the highest integer t such that f is t-th order correlation-immune.

In a combination generator, the correlation-immunity order t of the combining function determines the minimal number of LFSRs which must be considered together in a correlation attack. Indeed, the keystream produced by the combination generator is then independent from the probability distribution of any set of t constituent LFSRs. The smallest number of LFSRs involved in a correlation attack is therefore $t+1$. But the correlation-immunity order of a Boolean function cannot be chosen as high as we wish: it is limited by the algebraic degree of the function as shown in the next proposition.

Proposition 1.24. *[Sie84] Let f be a Boolean function of n variables. Then its correlation-immunity order t satisfies*

$$t + \deg f \leq n \; .$$

Moreover, if f is balanced and $t < n-1$, then

$$t + \deg f \leq n - 1 \; .$$

Proof. Let $u \in \mathbb{F}_2^n$ such that $wt(u) \geq n-t$. We compute the coefficients of the ANF of f with Theorem 1.3. For $L = \{1, \ldots, n\} \setminus \mathsf{Supp}(u)$, we have

$$a_u = \sum_{x_i = 0, i \in L} f(x_1, \ldots, x_n) \bmod 2 = 2^{-(n - wt(u))} wt(f) \bmod 2$$

where the last equality comes from the fact that the probability distribution of the output of f is unchanged when the $(n - wt(u)) \leq t$ variables defined by L are set to 0. If f is balanced, i.e., $wt(f) = 2^{n-1}$, we get that, for all u with $wt(u) \geq n-t$,

$$a_u = 2^{wt(u)-1} \bmod 2 = 0$$

since $wt(u) - 1 \geq n - t - 1 > 0$. In other words, all coefficients of degree greater than or equal to $(n-t)$ in the ANF of f are equal to zero, which means than $\deg f < n - t$.

If f is not balanced, we select some word $u^\star$ of weight $(n-t)$. Then,

$$a_{u^\star} = 2^{-t} wt(f) \bmod 2$$

implying that $wt(f) = 2^t a_{u^\star} + \Lambda 2^{t+1}$ for some integer Λ. Now, for any u of weight $(n - t + w)$ with $w \geq 1$, we get

$$a_u = 2^{-t+w} wt(f) = 2^w a_{u^\star} + \Lambda 2^{w+1} = 0 \bmod 2 \; .$$

This means that all coefficients in the ANF of degree greater than or equal to $(n - t + 1)$ vanish, i.e. $\deg f \leq n - t$. ◇

The correlation-immunity order of a Boolean function can be characterized by its Walsh transform. To this end, we need the following lemma which provides a very useful link between the Walsh transform and the restrictions of a function to some subspace.

Definition 1.25 (Restriction of a Boolean function)**.** *Let f be a Boolean function of n variables, and V be a linear subspace of $\mathbb{F}_2^n$ of dimension k. Let W be the supplementary space of V with respect to $\mathbb{F}_2^n$ i.e., V and W are in direct sum.*

Then, for any $w \in W$, the restriction of f to $w+V$ is the Boolean function of k variables, denoted by f_{w+V}, defined by

$$f_{w+V} : x \in V \times W \mapsto f(x+w)$$

The notion of restriction is of particular interest when V is the linear space spanned by some vectors of the canonical basis, i.e., $V = \langle e_i, i \in I \rangle$ for some $I \subset \{1, \ldots, n\}$, where e_i is the vector with support $\{i\}$. In this case, the restriction of f to a coset of V corresponds to f when the input variables $x_i, i \notin I$ are fixed.

Lemma 1.26 (Walsh transform and restriction)**.** *Let f be a Boolean function of n variables. Let V be a linear subspace of $\mathbb{F}_2^n$. Then, for any a, we have*

$$\sum_{v \in V} (-1)^{a \cdot v} \mathcal{E}(f + \varphi_v) = 2^{\dim V} \mathcal{E}(f_{a+V^\perp}) \ .$$

Proof.

$$\begin{aligned} \sum_{v \in V} (-1)^{a \cdot v} \mathcal{E}(f + \varphi_v) &= \sum_{v \in V} (-1)^{a \cdot v} \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) + v \cdot x} \\ &= \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)} \sum_{v \in V} (-1)^{(a+x) \cdot v} \\ &= 2^{\dim V} \sum_{x \in a + V^\perp} (-1)^{f(x)} \end{aligned}$$

where the last equality comes from the fact that, for any linear space $V \subset \mathbb{F}_2^n$,

$$\sum_{v \in V} (-1)^{a \cdot v} = \#\{v \in V : v \in \langle a \rangle^\perp\} - \#\{v \in V : v \notin \langle a \rangle^\perp\} = \begin{cases} 2^{\dim V} & \text{if } a \in V^\perp \\ 0 & \text{otherwise.} \end{cases}$$

◇

Proposition 1.27 ([XM88])**.** *Let f be a Boolean function of n variables. Then, f is t-th order correlation-immune if and only if $\mathcal{E}(F + \varphi_\alpha) = 0$ for any $\alpha \in \mathbb{F}_2^n$ with $1 \leq wt(\alpha) \leq t$.*

Proof. By definition, f is t-th order correlation-immune if and only if, for any $0 \leq k \leq t$ and any set $I \subset \{1, \ldots n\}$ of size k, the restriction of f to $a + \langle e_i, i \notin I \rangle$ has the same probability distribution as f:

$$2^{-(n-k)} \mathcal{E}(f_{a + \langle e_i, i \notin I \rangle}) = 2^{-n} \mathcal{E}(f) \ .$$

Let $V = \langle e_i, i \notin I \rangle$. We deduce from the previous lemma that this is equivalent to

$$\mathcal{E}(f) = 2^k \mathcal{E}(f_{a+V}) = \sum_{v \in V^\perp} (-1)^{a \cdot v} \mathcal{E}(f + \varphi_v) \ ,$$

This condition is obviously satisfied if all $\mathcal{E}(f+\varphi_v)=0$ for $v \in \langle e_i, i\in I\rangle \setminus \{0\}$. Conversely, we get by the inverse Walsh transform that,

$$\begin{aligned} 2^k \sum_{a\in V^\perp} (-1)^{a\cdot b}\mathcal{E}(f_{a+V}) &= \sum_{a\in V^\perp}\sum_{v\in V^\perp} (-1)^{a\cdot(b+v)}\mathcal{E}(f+\varphi_v) \\ &= \sum_{v\in V^\perp} \mathcal{E}(f+\varphi_v)\left(\sum_{a\in V^\perp}(-1)^{a\cdot(b+v)}\right) \\ &= 2^{n-k}\mathcal{E}(f+\varphi_b), \end{aligned}$$

using that, for $V=\langle e_i, i\notin I\rangle$, we have $V\cap V^\perp=\{0\}$. Therefore, if all restrictions satisfy $2^k\mathcal{E}(f_{a+V})=\mathcal{E}(f)$, then, for any nonzero $b\in\langle e_i, i\in I\rangle$,

$$2^{n-k}\mathcal{E}(f+\varphi_b)=\sum_{a\in V^\perp}(-1)^{a\cdot b}\mathcal{E}(f)=0\,.$$

◇

1.5.2 Approximation of a function by a function of $(t+1)$ variables.

Since the combining function in a combination generator is balanced and nonlinear, its correlation-immunity order is at most $(n-3)$. It follows that we can always apply a correlation attack by considering at most $(n-2)$ LFSRs together. We now show how to determine the best approximation of f by a Boolean function g depending on a fixed subset of k variables.

Proposition 1.28. *[Can02, Zha00] Let f be a function of n variables and let V be the subspace spanned by the k elements of the canonical basis of indices in $I=\{i_1,\ldots,i_k\}$. The highest possible value of $\mathcal{E}(f+g)$, over all k-variable functions g depending on $x_{i_1},\ldots,x_{i_k}$, is achieved if and only if*

$$\begin{cases} g(x)=1 & \text{if } \mathcal{E}(f_{x+V^\perp})<0 \\ g(x)=0 & \text{if } \mathcal{E}(f_{x+V^\perp})>0 \end{cases}$$

Moreover,

$$\max_g \mathcal{E}(f+g)=\sum_{x\in V}\left|\mathcal{E}(f_{x+V^\perp})\right|\ .$$

Proof. Using that V and $V^\perp$ are in direct sum, we have that, for any function g defined over V,

$$\begin{aligned} \mathcal{E}(f+g) &= \sum_{x\in V}\sum_{y\in V^\perp}(-1)^{f(x+y)+g(x)} \\ &= \sum_{x\in V}(-1)^{g(x)}\sum_{y\in V^\perp}(-1)^{f(x+y)} \\ &= \sum_{x\in V}(-1)^{g(x)}\mathcal{E}(f_{x+V^\perp})\,. \end{aligned}$$

It follows that $\mathcal{E}(f+g)$ is maximal if and only if all terms in the sum are greater than or equal to zero, or equivalently

$$g(x)=\begin{cases} 0 & \text{if } \mathcal{E}(f_{x+V^\perp})>0, \\ 1 & \text{if } \mathcal{E}(f_{x+V^\perp})<0. \end{cases} \tag{1.1}$$

Note that the value of $g(x)$ can be arbitrarily chosen when $\mathcal{E}(f_{x+V^\perp})=0$. The maximal value of $\mathcal{E}(f+g)$ directly follows. ◇

The previous proposition obviously implies that, for a balanced function f, the maximal value of $\mathcal{E}(f+g)$ is 0 if k is less than or equal to the correlation-immunity order of f since all $\mathcal{E}(f_{x+V^\perp}) = 0$ in this case. Another consequence is that, for $k = t+1$ where t is the correlation-immunity order of f, the maximal value of $\mathcal{E}(f+g)$ is achieved by an affine function.

Theorem 1.29. *[CT00] Let f be a t-resilient function of n variables and let V be the subspace spanned by the $(t+1)$ elements of the canonical basis of indices in $I = \{i_1, \ldots, i_{t+1}\}$. The highest possible value of $\mathcal{E}(f+g)$, over all $(t+1)$-variable functions g depending on $x_{i_1}, \ldots, x_{i_{t+1}}$, is achieved by the affine function*

$$g(x_{i_1}, \ldots, x_{i_{t+1}}) = \sum_{i \in I} x_i + \varepsilon$$

with $\varepsilon \in \mathbb{F}_2$.

Proof. We use the same notation as in the previous proposition. We have proved that, for any g defined over V

$$\mathcal{E}(f+g) = \sum_{x \in V} (-1)^{g(x)} \mathcal{E}(f_{x+V^\perp}) \; .$$

But, we know from Lemma 1.26 that

$$\mathcal{E}(f_{x+V^\perp}) = 2^{-(t+1)} \sum_{v \in V} (-1)^{x \cdot v} \mathcal{E}(f + \varphi_v) \; .$$

Moreover, since f is t-resilient, all its Walsh coefficients $\mathcal{E}(f+\varphi_v)$ for all v of weight at most t vanish. Then, the only term remaining in the previous sum is when v equals the all-one vector in V, i.e., the vector 1_I having I as support. We deduce

$$\mathcal{E}(f+g) = 2^{-(t+1)} \sum_{x \in V} (-1)^{g(x)} (-1)^{1_I \cdot x} \mathcal{E}(f + \varphi_{1_I}) = 2^{-(t+1)} \mathcal{E}(f + \varphi_{1_I}) \mathcal{E}(g + \varphi_{1_I}) \; .$$

This quantity is then maximized when $g = \varphi_{1_I} + \varepsilon$, and we get

$$\mathcal{E}(f+g) = \pm \mathcal{E}(f + \varphi_{1_I}) \; .$$

◇

This result is of great interest in the context of fast correlation attacks against combination generators. The correlation-immunity order of a Boolean function f can also be characterized by a combinatorial property of the code formed by the support of f, i.e., by $f^{-1}(1)$, namely by the so-called dual distance of this code [Del73, SM95].

Chapter 2

Cryptographic Sboxes

Many primitives like block ciphers use as building blocks some Boolean functions with several output bits. Such functions are called *vectorial Boolean functions*, or *S(ubstitution)-boxes*, by analogy with the former block-cipher standard DES.

2.1 Representations of an Sbox

2.1.1 Components of an Sbox

An Sbox from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ (i.e., with n inputs and m outputs) is usually represented by a collection of m Boolean functions of n variables called the *coordinates of f*.

Example 2.1 (A 4-bit Sbox)**.** The following table defines an Sbox from $\mathbb{F}_2^4$ into $\mathbb{F}_2^4$ (with hexadecimal notation).

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	f	e	b	c	6	d	7	8	0	3	9	a	4	2	1	5
$S_1(x)$	1	0	1	0	0	1	1	0	0	1	1	0	0	0	1	1
$S_2(x)$	1	1	1	0	1	0	1	0	0	1	0	1	0	1	0	0
$S_3(x)$	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	1
$S_4(x)$	1	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0

Table 2.1: Example of a 4-bit Sbox.

The Möbius transform enables us to compute the ANF of the four coordinates of this Sbox:

$$\begin{aligned}
S_1 &= 1 + x_1 + x_3 + x_2x_3 + x_4 + x_2x_4 + x_3x_4 + x_1x_3x_4 + x_2x_3x_4 \\
S_2 &= 1 + x_1x_2 + x_1x_3 + x_1x_2x_3 + x_4 + x_1x_4 + x_1x_2x_4 + x_1x_3x_4 \\
S_3 &= 1 + x_2 + x_1x_2 + x_2x_3 + x_4 + x_2x_4 + x_1x_2x_4 + x_3x_4 + x_1x_3x_4 \\
S_4 &= 1 + x_3 + x_1x_3 + x_4 + x_2x_4 + x_3x_4 + x_1x_3x_4 + x_2x_3x_4
\end{aligned}$$

Besides the coordinates, all linear combinations of the coordinates are usually involved for determining the cryptographic properties of an Sbox, in the sense of the following definition.

Definition 2.1. *Let S be an Sbox from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. The* Boolean components *of S are the n-variable Boolean functions*

$$S_\lambda : x \mapsto \lambda \cdot S(x)$$

for any $\lambda \in \mathbb{F}_2^m$. The component corresponding to $\lambda = 0$ is called the zero (or trivial) component.

In particular, the fact that an Sbox is invertible (i.e., a permutation) can be characterized by its coordinates.

Proposition 2.2. *Let S be an Sbox from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. S is a permutation if and only if all its non-trivial components are balanced.*

Proof. Let $\lambda \in \mathbb{F}_2^n \setminus \{0\}$. Then, if S is a permutation,

$$\mathcal{E}(S_\lambda) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\lambda \cdot S(x)} = \sum_{y \in \mathbb{F}_2^n} (-1)^{\lambda \cdot y} = 0 \, .$$

Conversely, for any $a \in \mathbb{F}_2^n$, we have

$$\begin{aligned} \sum_{\lambda \in \mathbb{F}_2^n} (-1)^{a \cdot \lambda} \mathcal{E}(S_\lambda) &= \sum_{\lambda \in \mathbb{F}_2^n} (-1)^{a \cdot \lambda} \sum_{x \in \mathbb{F}_2^n} (-1)^{\lambda \cdot S(x)} \\ &= \sum_{x \in \mathbb{F}_2^n} \left(\sum_{\lambda \in \mathbb{F}_2^n} (-1)^{(a + S(x)) \cdot \lambda} \right) \\ &= 2^n \#\{x \in \mathbb{F}_2^n : S(x) = a\} \, . \end{aligned}$$

It follows that, if all S_λ, $\lambda \neq 0$ are balanced, then for any $a \in \mathbb{F}_2^n$,

$$2^n \#\{x \in \mathbb{F}_2^n : S(x) = a\} \; = \mathcal{E}(S_0) = 2^n \, ,$$

i.e., S is a permutation. ◇

The degree of an Sbox is defined as the maximal degree of its Boolean components, since this is the most relevant notion from a mathematical point of view. The minimal degree of a non-trivial component may also be of interest in some attacks, but it won't be considered here.

Definition 2.3 (Degree of an Sbox)**.** *Let S be an Sbox from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. The degree of S is the highest degree of the ANF of its components.*

2.1.2 Univariate representation

When representing an n-bit Sbox, it may be convenient to identify the vector space $\mathbb{F}_2^n$ with the finite field with 2^n elements, $\mathbb{F}_{2^n}$. Then, for any fixed basis of $\mathbb{F}_2^n$ defining an isomorphism between $\mathbb{F}_2^n$ and $\mathbb{F}_{2^n}$, the n-bit Sbox S can be uniquely represented as a univariate polynomial in $\mathbb{F}_{2^n}[X]$:

$$S(X) = \sum_{i=0}^{2^n - 1} A_i X^i \, , A_i \in \mathbf{F}_{2^n} .$$

The coefficients of this polynomial are computed from the values of F by the discrete Fourier Transform (DFT) of F (aka Mattson-Solomon transform) (see e.g. [Bla83, MS77, GG04]).

Proposition 2.4 (Discrete Fourier transform of a function)**.** *Let S be a function from $\mathbb{F}_{2^n}$ into $\mathbb{F}_{2^n}$. Then, there exists a unique univariate polynomial in $\mathbb{F}_{2^n}[X]/(X^{2^n}+X)$ such that*

$$S(X) = \sum_{i=0}^{2^n-1} A_i X^i \ .$$

Moreover, $A_0 = S(0)$, $A_{2^n-1} = \sum_{x\in\mathbb{F}_{2^n}} S(x)$ and the coefficients A_i, $1 \le i \le 2^n-2$, are given by the discrete Fourier transform of the values of S at all nonzero inputs, namely

$$A_i = \sum_{k=0}^{2^n-2} S(\alpha^k)\alpha^{-ki},\ 0 \le i \le 2^n-2$$

where α is a primitive element in $\mathbb{F}_{2^n}$.

Proof. We first prove that the univariate polynomial

$$A(X) = \sum_{i=0}^{2^n-1} A_i X^i$$

with the previously defined coefficients takes the same values as S. Obviously,

$$A(0) = A_0 = S(0) \ .$$

Now, for any nonzero element α^j, we have

$$A(\alpha^j) = \sum_{i=0}^{2^n-1} A_i \alpha^{ij} = (A_0 + A_{2^n-1}) + \sum_{i=1}^{2^n-2} A_i \alpha^{ij} \ .$$

By definition, the constant coefficient of this polynomial is then

$$A_0 + A_{2^n-1} = \sum_{k=0}^{2^n-1} S(\alpha^k) \ .$$

It then follows that

$$\begin{aligned} A(\alpha^j) &= \sum_{k=0}^{2^n-1} S(\alpha^k) + \sum_{i=1}^{2^n-2} \left(\sum_{k=0}^{2^n-2} S(\alpha^k)\alpha^{-ki} \right) \alpha^{ij} \\ &= \sum_{i=0}^{2^n-2} \left(\sum_{k=0}^{2^n-2} S(\alpha^k)\alpha^{-ki} \right) \alpha^{ij} \\ &= \sum_{k=0}^{2^n-2} S(\alpha^k) \left(\sum_{i=0}^{2^n-2} \alpha^{i(j-k)} \right) = S(\alpha^j) \ . \end{aligned}$$

Then, the values of A correspond to the values of S. Moreover, this polynomial is unique. Otherwise, there would exist two distinct polynomials of degree less than or equal to (2^n-1) taking the same value at 2^n distinct points, which is impossible. ◇

Example 2.2. Let α be a root of the irreducible polynomial $1 + x + x^4$. We identify $\mathbb{F}_2^4$ with $\mathbb{F}_{2^4}$ using the basis $\{1, \alpha, \alpha^2, \alpha^3\}$. This leads to

$\mathbf{F}_{2^4}$	0	1	α	α^2	α^3	α^4	α^5	α^6	α^7
	0	1	α	α^2	α^3	$\alpha+1$	$\alpha^2+\alpha$	$\alpha^3+\alpha^2$	$\alpha^3+\alpha+1$
$\mathbf{F}_2^4$	0000	0001	0010	0100	1000	0011	0110	1100	1011

α^8	α^9	α^{10}	α^{11}	α^{12}	α^{13}	α^{14}
α^2+1	$\alpha^3+\alpha$	$\alpha^2+\alpha+1$	$\alpha^3+\alpha^2+\alpha$	$\alpha^3+\alpha^2+\alpha+1$	$\alpha^3+\alpha^2+1$	α^3+1
0101	1010	0111	1110	1111	1101	1001

Considering the following Sbox S over $\mathbb{F}_{2^4}$,

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	f	e	b	c	6	d	7	8	0	3	9	a	4	2	1	5

we obtain the univariate representation:

$$\begin{aligned} S(X) &= \alpha^{12} + \alpha^2 X + \alpha^{13} X^2 + \alpha^6 X^3 + \alpha^{10} X^4 + \alpha X^5 + \alpha^{10} X^6 + \alpha^2 X^7 \\ &\quad + \alpha^9 X^8 + \alpha^4 X^9 + \alpha^7 X^{10} + \alpha^7 X^{11} + \alpha^5 X^{12} + X^{13} + \alpha^6 X^{14} \end{aligned}$$

It is worth noticing that the univariate representation of an Sbox depends on the basis chosen for identifying $\mathbb{F}_2^n$ and $\mathbb{F}_{2^n}$.

A particular family of Sboxes is the class of functions whose univariate representation contains a single monomial (up to a change of basis), since they usually have a lower implementation cost. These functions are called *monomial* or *power* functions. The most prominent example in this family is the AES Sbox.

Example 2.3. The AES Sbox, which operates in parallel on each byte of the input of the round function is defined as a permutation of the finite field with 2^8 elements. The isomorphism between the field $\mathbb{F}_{2^8}$ and the vector space $\mathbb{F}_2^8$ is given by

$$(x_0, x_1, \ldots, x_7) \in \mathbf{F}_2^8 \mapsto \sum_{i=0}^{7} x_i X^i$$

where all operations are modulo the irreducible polynomial

$$X^8 + X^4 + X^3 + X + 1.$$

With this identification, the AES Sbox corresponds to the inversion in $\mathbb{F}_{2^8}$ (where the inverse of 0 is 0), i.e.,

$$x \in \mathbf{F}_{2^8} \mapsto x^{254},$$

followed by an affine function over $\mathbb{F}_2^8$:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

Some properties of the univariate representation of an Sbox are related to the ANF of its components. For instance, the degree of S can be deduced from the degree of its univariate representation.

Proposition 2.5. *Let S be an n-bit Sbox and let*

$$\sum_{i=0}^{2^n-1} A_i X^i$$

denote its univariate representation in $\mathbb{F}_{2^n}[X]$. Then,

$$\deg(f) = \max\{wt(i) : 0 \leq i < 2^n \text{ and } A_i \neq 0 \}$$

Proof. Let $\{\alpha_0, \ldots, \alpha_{n-1}\}$ be a basis of $\mathbb{F}_{2^n}$. We first prove that the degree of S cannot exceed the maximum weight of an exponent i with $A_i \neq 0$. Indeed, the multivariate representation of the monomial X^s over $\mathbb{F}_{2^n}$ is

$$\begin{aligned} X^s &= (\alpha_0 x_0 + \alpha_1 x_1 + \ldots + \alpha_{n-1} x_{n-1})^s \\ &= (\alpha_0 x_0 + \alpha_1 x_1 + \ldots + \alpha_{n-1} x_{n-1})^{\sum_{i=0}^{n-1} s_i 2^i} \\ &= \prod_{i=0}^{n-1} \left(\alpha_0^{2^i} x_0^{2^i} + \alpha_1^{2^i} x_1^{2^i} + \ldots + \alpha_{n-1}^{2^i} x_{n-1}^{2^i} \right)^{s_i} . \end{aligned}$$

Then, we get a product of $wt(s)$ terms of degree 1 implying that the degree of X^s cannot exceed $wt(s)$.

Conversely, we observe that the number of n-variable Boolean functions of degree at most d is $2^{\dim \mathcal{R}(d,n)}$ with

$$\dim \mathcal{R}(d,n) = \sum_{i=0}^{d} \binom{n}{i} ,$$

implying that the number of n-bit Sboxes with degree d is

$$2^{n \dim \mathcal{R}(d,n)} .$$

On the other hand, the number of univariate polynomials composed of monomials X^s with $wt(s) \leq d$ is also

$$(2^n)^{\sum_{i=0}^{d} \binom{n}{i}} = 2^{n \dim \mathcal{R}(d,n)} .$$

It follows that the two sets of functions are the same. ◇

2.2 Exploiting the degree of the Sbox

2.2.1 Basic algebraic attack

The basic principle of algebraic attacks goes back to Shannon's work [Sha49, Page 711]: these techniques consist in expressing the whole cipher as a large system of multivariate algebraic equations. In a known-plaintext attack, the plaintext and ciphertext bits are replaced by their values and the unknowns correspond to the key bits. The secret key can then be recovered by solving this algebraic system. Therefore, a major parameter which influences the complexity of such an attack is the degree of the underlying system.

A naive method for solving such a system of degree d is the *linearization*. It consists in identifying the system with a linear system of $\sum_{i=1}^{d} \binom{n}{i}$ variables, where each product of i initial variables ($1 \leq i \leq d$) is seen as a new variable. The solution is then found by a Gaussian reduction (or by more sophisticated techniques) whose time complexity is roughly

$$\left(\sum_{i=1}^{d} \binom{n}{i}\right)^{\omega} \simeq n^{\omega d} ,$$

where ω is the exponent of the matrix inversion algorithm, i.e., $\omega \simeq 2.37$ [CW90]. Some much more sophisticated techniques exist for solving polynomial systems over $\mathbb{F}_2$. Actually, this problem has been extensively studied in computer algebra and it is well-known that some methods based on Gröbner basis algorithms efficiently apply, see e.g. [Fau99, Ste04, Fau02]. Some ad-hoc techniques including XL [CKPS00] or XSL [CP02] have also been proposed. See e.g. [Cid04] and [CAA+08] for a discussion on these algorithms.

Example 2.4. (inspired from [KR11]). Let us consider a toy-cipher composed of a single-round key-alternating cipher which operates on 4-bit inputs under an 8-bit key (K_0, K_1).

$$m \longrightarrow \overset{K_0}{\underset{\downarrow}{\oplus}} \longrightarrow u \longrightarrow \boxed{S} \longrightarrow v \longrightarrow \overset{K_1}{\underset{\downarrow}{\oplus}} \longrightarrow c$$

The inner permutation of $\mathbb{F}_2^4$ is defined as in Table 2.1.

Using the algebraic expressions of its coordinates, we can now express each ciphertext bit c_i, $1 \leq i \leq 4$, as a multivariate polynomial in the plaintext bits $m_1, \ldots, m_4$ and in the key bits $k_1, \ldots, k_8$:

$$\begin{aligned}
c_1 + k_5 &= 1 + (m_1 + k_1) + (m_3 + k_3) + (m_2 + k_2)(m_3 + k_3) + (m_4 + k_4) + (m_2 + k_2)(m_4 + k_4) \\
&\quad + (m_3 + k_3)(m_4 + k_4) + (m_1 + k_1)(m_3 + k_3)(m_4 + k_4) + (m_2 + k_2)(m_3 + k_3)(m_4 + k_4) \\
c_2 + k_6 &= 1 + (m_1 + k_1)(m_2 + k_2) + (m_1 + k_1)(m_3 + k_3) + (m_1 + k_1)(m_2 + k_2)(m_3 + k_3) + (m_4 + k_4) \\
&\quad + (m_1 + k_1)(m_4 + k_4) + (m_1 + k_1)(m_2 + k_2)(m_4 + k_4) + (m_1 + k_1)(m_3 + k_3)(m_4 + k_4) \\
c_3 + k_7 &= 1 + (m_2 + k_2) + (m_1 + k_1)(m_2 + k_2) + (m_2 + k_2)(m_3 + k_3) + (m_4 + k_4) + (m_2 + k_2)(m_4 + k_4) \\
&\quad + (m_1 + k_1)(m_2 + k_2)(m_4 + k_4) + (m_3 + k_3)(m_4 + k_4) + (m_1 + k_1)(m_3 + k_3)(m_4 + k_4) \\
c_4 + k_8 &= 1 + (m_3 + k_3) + (m_1 + k_1)(m_3 + k_3) + (m_4 + k_4) + (m_2 + k_2)(m_4 + k_4) + (m_3 + k_3)(m_4 + k_4) \\
&\quad + (m_1 + k_1)(m_3 + k_3)(m_4 + k_4) + (m_2 + k_2)(m_3 + k_3)(m_4 + k_4)
\end{aligned}$$

Every plaintext-ciphertext pair then provides four equations in the eight key-bits:

$$\begin{aligned}
c_1 + k_5 &= S_1(m) + (1 + m_3m_4)k_1 + (m_3 + m_4 + m_3m_4)k_2 + (1 + m_2 + m_4 + m_1m_4 + m_2m_4)k_3 \\
&\quad +(1 + m_2 + m_3 + m_1m_3 + m_2m_3)k_4 + m_4k_1k_3 + m_3k_1k_4 + (1 + m_4)k_2k_3 + (1 + m_3)k_2k_4 \\
&\quad +(1 + m_1 + m_2)k_3k_4 + k_1k_3k_4 + k_2k_3k_4 \\
c_2 + k_6 &= S_2(m) + (m_2 + m_3 + m_2m_3 + m_4 + m_2m_4 + m_3m_4)k_1 + (m_1 + m_1m_3 + m_1m_4)k_2 \\
&\quad +(m_1 + m_1m_2 + m_1m_4)k_3 + (1 + m_1 + m_1m_2 + m_1m_3)k_4 + (1 + m_3 + m_4)k_1k_2 \\
&\quad +(1 + m_2 + m_4)k_1k_3 + (1 + m_2 + m_3)k_1k_4 + m_1k_2k_3 + m_1k_2k_4 + m_1k_3k_4 + k_1k_2k_3 \\
&\quad +k_1k_2k_4 + k_1k_3k_4 \\
c_3 + k_7 &= S_3(m) + (m_2 + m_2m_4 + m_3m_4)k_1 + (1 + m_1 + m_3 + m_4 + m_1m_4)k_2 \\
&\quad +(m_2 + m_4 + m_1m_4)k_3 + (1 + m_2 + m_3 + m_1m_2 + m_1m_3)k_4 + (1 + m_4)k_1k_2 \\
&\quad +m_4k_1k_3 + (m_2 + m_3)k_1k_4 + k_2k_3 + m_1k_3k_4 + (1 + m_1)k_2k_4 + k_3k_4 + k_1k_2k_4 + k_1k_3k_4 \\
c_4 + k_8 &= S_4(m) + (m_3 + m_3m_4)k_1 + (m_4 + m_3m_4)k_2 + (1 + m_1 + m_4 + m_1m_4 + m_2m_4)k_3 \\
&\quad +(1 + m_2 + m_3 + m_1m_3 + m_2m_3)k_4 + (1 + m_4)k_1k_3 + (m_3)k_1k_4 + m_4k_2k_3 \\
&\quad +(1 + m_3)k_2k_4 + (1 + m_1 + m_2)k_3k_4 + k_1k_3k_4 + k_2k_3k_4
\end{aligned}$$

For instance, the knowledge of $E(0) = \texttt{0x4}$ leads to

$$\begin{aligned}
c_1 + k_5 &= 1 + k_1 + k_3 + k_4 + k_2k_3 + k_2k_4 + k_3k_4 + k_1k_3k_4 + k_2k_3k_4 \\
c_2 + k_6 &= 1 + k_4 + k_1k_2 + k_1k_3 + k_1k_4 + k_1k_2k_3 + k_1k_2k_4 + k_1k_3k_4 \\
c_3 + k_7 &= 1 + k_2 + k_4 + k_1k_2 + k_2k_3 + k_2k_4 + k_3k_4 + k_1k_2k_4 + k_1k_3k_4 \\
c_4 + k_8 &= 1 + k_3 + k_4 + k_1k_3 + k_2k_4 + k_3k_4 + k_1k_3k_4 + k_2k_3k_4
\end{aligned}$$

By collecting such equations, we get a polynomial system of degree 3 with 8 unknowns. Then, we can linearize the system by identifying each monomial in the key bits of degree 2 or 3 with a new unknown:

$$k_9 = k_1k_2,\ \ k_{10} = k_1k_3, \ldots\ k_{14} = k_3k_4,\ \ k_{15} = k_1k_2k_3, \ldots, k_{18} = k_2k_3k_4$$

transforming the previous four equations into

$$\begin{aligned}
c_1 + k_5 &= 1 + k_1 + k_3 + k_4 + k_{12} + k_{13} + k_{14} + k_{16} + k_{18} \\
c_2 + k_6 &= 1 + k_4 + k_9 + k_{10} + k_{11} + k_{15} + k_{17} + k_{16} \\
c_3 + k_7 &= 1 + k_2 + k_4 + k_9 + k_{12} + k_{13} + k_{14} + k_{17} + k_{16} \\
c_4 + k_8 &= 1 + k_3 + k_4 + k_{10} + k_{13} + k_{14} + k_{16} + k_{18}
\end{aligned}$$

This way, we get a linear system with $8 + \binom{4}{2} + \binom{4}{3} = 18$ unknowns, which can be solved as far as it has enough equations, i.e., as far as 5 plaintext-ciphertext pairs are known, leading to 20 equations.

Practical block ciphers have obviously more than a single round. In this case, the degree of the polynomial system is expected to grow as $(\deg S)^r$ when the number of rounds r increases. Solving such a system then becomes infeasible even for a few rounds only and with sophisticated techniques. An alternative solution consists in using some intermediate variables in order to handle equations of a reasonable degree. For instance, let us consider the following

two-round key-alternating cipher

$$\begin{array}{ccccccccccccc} & & K_0 & & & & K_1 & & & & K_2 & & \\ & & \downarrow & & & & \downarrow & & & & \downarrow & & \\ m \to & \oplus & \to u \to & \boxed{S} & \to v \to & \oplus & \to w \to & \boxed{S} & \to x \to & \oplus & \to c\,. \end{array}$$

Then, we can consider the 4 bits of the intermediate value v as four additional unknowns. With this method, a plaintext-ciphertext pair provides 8 equations of degree 3 involving 16 unknowns (the twelve key-bits and the four bits of v). Any additional plaintext-ciphertext pair provides four new equations, but introduces four more unknowns. Therefore, from N plaintext-ciphertext pairs, we obtain a system with $8N$ equations and $(12 + 4N)$ unknowns.

2.2.2 Enhanced algebraic attack

Courtois and Pieprzyk [CP02] have pointed out that it might be possible to lower the degree of the polynomial system that we need to solve even if the round function has a high degree. Indeed, the attacker may equivalently exploit any Boolean relation between the plaintext bits, the ciphertext bits and the key-bits.

Example 2.5. If we come back to our previous toy-cipher, we can check that, even if the inner permutation S has degree 3, there exist some Boolean relations of degree 2 only between its inputs and outputs, for instance it can be checked that

$$x_2x_4 + x_2S_1(x_1, \ldots, x_4) + x_2S_2(x_1, \ldots, x_4) = 0$$

for all inputs. Any plaintext-ciphertext pair for our single-round toy-cipher then leads to the following quadratic equation

$$(m_4 + c_1 + c_2)k_2 + m_2k_4 + m_2k_5 + m_2k_6 + k_2k_4 + k_2k_5 + k_2k_6 = m_2m_4 + m_2c_1 + m_2c_2\,.$$

A total of 21 linearly independent relations of degree of 2 between the input and output bits of S can be exhibited, implying that the derived polynomial system is easier to solve than the original cipher equations.

An important quantity which affects the complexity of this algebraic attack is obviously the lowest degree we can reach for a Boolean relation between the inputs and outputs of the round permutation.

Proposition 2.6. *Let S be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. The Boolean relations of degree at most d between the inputs and outputs of S are the elements of the kernel of the binary matrix with $\sum_{i=0}^{d}\binom{2n}{i}$ rows and 2^n columns whose rows correspond to the value vectors of the n-variable Boolean function*

$$x^u S(x)^v, u, v \in \mathbb{F}_2^n \text{ such that } wt(u) + wt(v) \leq d\,.$$

Most notably, the number of linearly independent relations of degree at most d is at least

$$\sum_{i=0}^{d}\binom{2n}{i} - 2^n\,.$$

Proof. This result comes from the fact that the relations between x and $S(x)$ correspond to the functions

$$\sum_{u,v\in\mathbb{F}_2^n} c_{u,v} x^u \left[S(x)\right]^v$$

which vanish for all possible inputs x. Moreover, the degree of such a relation corresponds to the maximal value of $wt(u) + wt(v)$ such that $c_{u,v} = 1$. Any such relation is then defined by a linear combination of the monomials $x^u S(x)^v$ which vanish at all points, i.e., an element in the kernel of the matrix defined by the value vectors of all these monomials. The involved matrix has 2^n columns, and $\sum_{i=0}^{d} \binom{2n}{i}$ rows (corresponding to all possible pairs (u, v)). Then, the dimension of its kernel exceeds the difference between the number of rows and the number of columns. ◇

We deduce for instance that any function from $\mathbb{F}_2^4$ into $\mathbb{F}_2^4$ has at least

$$\sum_{i=0}^{2} \binom{8}{i} - 2^4 = 37 - 16 = 21$$

quadratic relations between its inputs and outputs, as we observed in the toy-example.

Case of the AES. The AES Sbox can be seen as the composition of the inversion over $\mathbb{F}_{2^8}$ with an affine function. More precisely,

$$S : A \circ \varphi^{-1} \circ (\varphi(x))^{254} \ ,$$

where φ is the isomorphism from $\mathbb{F}_2^8$ into $\mathbb{F}_{2^8}$ defined by the basis $\{1, \alpha, \ldots, \alpha^7\}$ and α is a root of $X^8 + X^4 + X^3 + X + 1$. By definition, the inversion Inv over $\mathbb{F}_{2^8}$ satisfies

$$x^2\, \mathsf{Inv}(x) = x^2 x^{254} = x \ .$$

Since $x \mapsto x^2$ is an $\mathbb{F}_2$-linear mapping over $\mathbb{F}_2^8$, we deduce that this relation corresponds to eight Boolean relations over $\mathbb{F}_2$ between the inputs and outputs of S. Actually, there exists 39 such quadratic relations for the AES Sbox. This is much higher than expected for a randomly chosen mapping over $\mathbb{F}_2^8$, since Proposition 2.6 shows that there is no relation of degree 3 for a random 8-bit Sbox.

Using these relations of degree 2, a quadratic system can be formed by introducing intermediate variables corresponding to the outputs of the successive rounds. However, solving this system is infeasible due to its size: 8000 quadratic equations of 1600 variables.

2.2.3 Other attacks exploiting a low degree

There are some other attacks exploiting that the Sbox has a low degree. A first example is the higher-order differential cryptanalysis [Knu95] which exploits the fact that the Boolean function corresponding to some ciphertext bit has a low degree. This property is then used as a distinguisher on the reduced cipher, using the same approach as classical statistical attacks like differential cryptanalysis, even if the distinguishing relation holds with probability 1. A nice example is the cryptanalysis of the $\mathcal{KN}$ cipher by Jakobsen and Knudsen [JK97]. The degree d of the Boolean function involved in the attack determines the data complexity of the cryptanalysis, which is equal to 2^{d+1} chosen plaintext-ciphertext pairs. Since higher-order

differential attacks on block ciphers usually exploit a low degree of several iterations of the round function, it requires a careful analysis of the growth of the degree when the function is iterated. This analysis depends on the general structure of the cipher (e.g. SPN or Feistel structure). Some results on SPN can be found in [BCD11] for instance. The more recent *cube attacks* [DS09] and *cube testers* [ADMS09] can be seen as variants of higher-order differential cryptanalysis.

A different type of attacks is the so-called *interpolation attack* introduced in [JK97]. It exploits the fact that the encryption function has a low *univariate degree*. For any given key, the univariate representation of the encryption (or decryption) function can then be recovered by Lagrange interpolation as soon as enough plaintext-ciphertext pairs are known. The data complexity is then determined by the univariate degree of the function. An interesting point is that this technique provides the attacker with a way to decrypt any ciphertext but does not recover the key.

2.3 Linear properties of Sboxes

2.3.1 Linear cryptanalysis

The principle of linear cryptanalysis has been originally presented by Gilbert, Chassé and Tardy-Corfdir [GC91, TCG91] on the block cipher FEAL, and then applied to DES by Matsui [Mat94, Mat95]. It exploits, as a distinguisher a linear Boolean relation between the input bits, output bits and the key-bits of the reduced cipher, which is biased, i.e., which holds with a probability different from 1/2. In other words, it uses a triple of *masks* $(\alpha, \beta, \gamma) \in \mathbb{F}_2^n \times \mathbb{F}_2^n \times \mathbb{F}_2^\kappa$ such that

$$\Pr_x[\alpha \cdot x + \beta \cdot G_k(x) + \gamma \cdot k = 0] = \frac{1}{2}(1 + \varepsilon) \text{ with } \varepsilon \neq 0 \, .$$

Example 2.6. We consider the same cipher with a 4-bit block and an 8-bit key as in Example 2.4. Recall that its inner permutation is defined by

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	f	e	b	c	6	d	7	8	0	3	9	a	4	2	1	5
$S_1(x)$	1	0	1	0	0	1	1	0	0	1	1	0	0	0	1	1
$S_2(x)$	1	1	1	0	1	0	1	0	0	1	0	1	0	1	0	0
$S_3(x)$	1	1	0	1	1	1	1	0	0	0	0	0	1	0	0	1
$S_4(x)$	1	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0

It can be checked that the linear relation defined by the input mask $\alpha = (1, 0, 0, 1) =$ `0x9` and $\beta = (0, 1, 0, 0) =$ `0x2` has the following value vector (with hexadecimal notation):

$$x_1 + x_4 + S_2(x) = \texttt{0x7ffd} \, .$$

In particular, its Hamming weight is 14 implying that this relation holds with probability $\frac{2}{16} = \frac{1}{8}$. From this biased relation, we can derive a distinguisher for the whole cipher described in Example 2.4:

$$(\alpha \cdot m) + (\beta \cdot c) = (\alpha \cdot K_0) + (\beta \cdot K_1) + 1$$

with probability 7/8. It recovers the key information-bit $(k_1 + k_4 + k_5 + k_8)$ by a simple majority vote between the binary values taken by $m_1 + m_4 + c_2 + 1$ for a few plaintext-ciphertext pairs.

2.3.2 Linearity of an Sbox

In a linear attack, we then search for the most biased linear relation between the input and output bits of the Sbox. For a pair of masks (a, b), the corresponding bias is determined by the Walsh coefficient of S_b at point a, $\mathcal{E}(S_b + \varphi_a)$. These biases are usually represented as a two-dimensional table $\mathcal{E}(a, b) = \mathcal{E}(S_b + \varphi_a)$ called the *linear-approximation table*.

Example 2.7. The linear-approximation table of the previous Sbox corresponds to the following table.

$a \setminus b$	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	-4	.	4	.	-4	8	-4	4	8	4	.	-4	.	4	.
2	4	-4	.	-4	.	.	4	4	8	.	4	8	-4	-4	.
3	8	4	4	-4	4	.	.	.	.	4	-4	-4	-4	.	8
4	.	-4	4	4	-4	.	.	-8	.	4	4	4	4	.	8
5	-4	4	.	4	8	.	4	-4	8	.	-4	.	4	-4	.
6	-4	.	4	.	4	8	4	4	-8	4	.	4	.	-4	.
7	.	.	.	8	.	-8	.	.	.	.	8	.	8	.	.
8	.	-4	4	-8	.	4	4	-8	.	-4	-4	.	.	4	-4
9	-4	-12	.	.	4	-4	.	4	.	.	-4	-4	.	.	4
a	-4	.	-12	-4	.	4	.	-4	.	4	.	.	-4	.	4
b	.	.	.	4	-4	4	-4	.	.	-8	-8	4	-4	-4	4
c	.	.	.	-4	-4	-4	-4	.	.	8	-8	4	4	-4	-4
d	-4	.	4	4	.	-4	.	-4	.	4	.	.	-12	.	-4
e	4	-4	.	.	4	4	-8	-4	.	.	4	-4	.	-8	-4
f	-8	4	4	-8	.	-4	-4	.	.	-4	4	.	.	-4	4

It can be checked that $\mathcal{E}(\texttt{0x9}, \texttt{0x2}) = -12$ as exploited in the previously described attack.

A major parameter is then the highest magnitude of an element in this table, which corresponds to the linearity of the Sbox.

Definition 2.7. *Let S be function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. Then, the* linearity *of S is the highest linearity of any of its non-trivial components, i.e.*

$$\mathcal{L}(S) = \max_{b \in (\mathbb{F}_2^m)^*} \mathcal{L}(S_b) = \max_{a,b \in (\mathbb{F}_2^m)^*} |\mathcal{E}(S_b + \varphi_a)| \, .$$

The linearity of S is now related to the weights of the following code.

Proposition 2.8 ([CCZ98])**.** *Let S be an Sbox from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. Let us consider the code Γ_S having for generator matrix the following $(2n+1) \times 2^n$-matrix*

$$\begin{pmatrix} S(0) & S(1) & S(2) & \dots & S(2^n-1) \\ 0 & 1 & 2 & \dots & 2^n-1 \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix}$$

where the integers in the first two rows of the matrix correspond to n-bit vectors, i.e., each element in these two rows is an n-bit column vector. The Hamming weights of this code are

$$\left\{ 2^{n-1} - \frac{1}{2}\mathcal{E}(S_b + \varphi_a); \;\; 2^{n-1} + \frac{1}{2}\mathcal{E}(S_b + \varphi_a), a, b \in \mathbb{F}_2^n \right\}$$

The proof is similar to the proof of Proposition 1.19. Note that the case a and b equal to zero should be included in order to take into account the words of $\mathcal{R}(1,n)$. Based on some relationships satisfied by the weight distribution of a code with such parameters, we can deduce a lower bound on the linearity of an Sbox.

Proposition 2.9. *[CV95, CCZ98] Let S be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. Then*

$$\mathcal{L}(S) \geq 2^{\frac{n+1}{2}} .$$

The Sboxes for which equality holds are called almost bent (AB) functions. *They exist for n odd only.*

When n is even, almost bent functions do not exist, and the lowest possible linearity for an n-bit Sbox is not known. But, the best known value is $\mathcal{L}(S) = 2^{\frac{n}{2}+1}$, and this value is tight for a very few families of Sboxes, including the inversion over $\mathbb{F}_{2^n}$ which is used in the AES.

2.4 Differential properties of Sboxes

2.4.1 Differential cryptanalysis

Differential cryptanalysis has been introduced by Biham and Shamir [BS91]. It exploits as a distinguishing property the existence of a *differential*, i.e., of a pair of differences (a,b) in $\mathbb{F}_2^n$ such that

$$\Pr_X[E_k(X+a) + E_k(X) = b]$$

is high. For a randomly chosen permutation, the derivative $X \mapsto E_k(X+a) + E_k(X)$ is expected to take every possible nonzero value with uniform probability. It follows that a differential can be used as a distinguisher if the corresponding probability p is significantly different from 2^{-n}.

2.4.2 Differential uniformity

Then, a relevant parameter for evaluating the resistance of a cipher against differential attacks is the maximal probability that a given nonzero input difference leads to a given output difference for the Sbox. This quantity is determined by the *differential uniformity* of S.

Definition 2.10 (Differential uniformity [Nyb93])**.** *Let S be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$. For any $a \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2^m$, we define*

$$\delta(a,b) = \#\{x \in \mathbb{F}_2^n \ : S(x+a) + S(x) = b\} .$$

Then the multi-set $\{\delta(a,b),\ a \in \mathbb{F}_2^n \setminus \{0\}, b \in \mathbb{F}_2^m\}$ is the differential spectrum *of S, and its maximum*

$$\delta_S = \max_{a \neq 0, b} \delta(a,b)$$

is the differential uniformity *of S.*

The values $(\delta(a,b))_{a \in \mathbb{F}_2^n, b \in \mathbb{F}_2^m}$ are usually represented as a two-dimensional array called the *difference table* of S.

Example 2.8. The difference table of the Sbox in Table 2.1 is as follows.

$a \setminus b$	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
1	2	0	4	2	0	2	2	0	0	0	2	0	0	0	2
2	2	2	0	2	4	0	2	0	4	0	0	0	0	0	0
3	2	0	4	0	2	0	0	0	0	6	0	0	0	2	0
4	2	0	2	4	0	0	0	2	2	0	0	2	0	0	2
5	0	4	2	0	0	0	2	2	0	0	4	2	0	0	0
6	4	0	0	0	0	4	0	4	0	0	0	0	4	0	0
7	0	2	0	0	2	2	2	0	2	2	2	0	0	2	0
8	0	4	0	0	0	4	0	0	0	0	0	0	4	0	4
9	2	2	0	2	2	0	0	0	4	0	0	2	0	2	0
a	0	0	2	2	0	2	2	2	0	2	2	0	0	0	2
b	0	0	2	0	4	0	2	2	0	0	0	6	0	0	0
c	0	2	0	0	0	2	0	0	2	2	2	2	0	4	0
d	2	0	0	0	2	0	0	0	0	2	0	0	8	2	0
e	0	0	0	0	0	0	4	0	0	0	4	0	0	4	4
f	0	0	0	4	0	0	0	4	2	2	0	2	0	0	2

The lowest differential uniformity that can be achieved by an Sbox is determined in the following proposition.

Proposition 2.11 ([NK93])**.** *Let S be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. Then, its differential uniformity satisfies*

$$\delta(S) \geq 2 \, .$$

The Sboxes for which equality holds are called almost perfect nonlinear (APN) *functions.*

Proof. The proof comes from the fact that

$$\delta(a, b) = \#\{x \in \mathbb{F}_2^n, S(x + a) + S(x) = b\}$$

is always even since, if x is a solution of this equation, $(x + a)$ is a solution too. ◇

The *APN* terminology comes from the fact that we call *perfect nonlinear* the functions from $\mathbb{F}_2^n$ into $\mathbb{F}_2^m$ such that $\delta(S) = 2^{n-m}$. It can be proved that the perfect nonlinear functions correspond to the functions with linearity $\mathcal{L}(S) = 2^{\frac{n}{2}}$, i.e., such that all their components are bent. Such functions exist only when $n \geq 2m$ [Nyb91]. Therefore, when $m = n$, the optimal differential uniformity is obtained for the almost perfect nonlinear functions.

2.5 Link between differential uniformity and linearity

The differential spectrum and the Walsh spectrum of an Sbox are strongly related. This link appears under several forms. We first focus on the coding approach.

We have seen in Proposition 2.8 that the Walsh spectrum of an Sbox is related to the weight distribution of the code Γ_S with generator matrix

$$G = \begin{pmatrix} S(0) & S(1) & S(2) & \dots & S(2^n - 1) \\ 0 & 1 & 2 & \dots & 2^n - 1 \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix} \tag{2.1}$$

Now, we show that the APN property is also related to this code.

Proposition 2.12 ([CCZ98])**.** *Let S be an Sbox from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. Let us consider the code Γ_S defined by the generator matrix in Eq. (2.1). Then, the minimum distance of $\Gamma_S^\perp$ is equal to 6 if S is APN and equal to 4 otherwise.*

Proof. It can be easily checked that the minimum distance of $\Gamma_S^\perp$ is even (since Γ_S contains the all-one codeword), and at least 4. Moreover, it cannot be greater than 6 [DZ84, BT93]. Now, a codeword of weight 4 of $\Gamma_S^\perp$ corresponds to four columns of the generator matrix G which sum to zero (since the words of $\Gamma_S^\perp$ are the column vectors c such that $Gc = 0$). Therefore, $\Gamma_S^\perp$ has minimum distance 4 if and only if there exist four distinct elements x_1, x_2, x_3, x_4 such that

$$x_1 + x_2 + x_3 + x_4 = 0 \text{ and } S(x_1) + S(x_2) + S(x_3) + S(x_4) = 0 \ .$$

Replacing x_2 by $x_1 + a$, we get that it is equivalent to the existence of x_1, x_3 and a such that

$$S(x_1) + S(x_1 + a) = S(x_3) + S(x_3 + a)$$

since $x_4 = x_3 + (x_1 + x_2) = x_3 + a$. All four x_i are distinct if and only if they correspond to four distinct solutions of the equation

$$S(x + a) + S(x) = b$$

with $b = S(x_1) + S(x_1 + a)$. This equivalently means that S is not APN. ◇

The APN and AB properties then correspond to a particular value of the minimum distance of the code $\Gamma_S^\perp$ and Γ_S. Using the relations between the weight distributions of the two codes, it can be deduced that these two properties are related as follows.

Proposition 2.13 ([CV95, CCZ98])**.** *Any almost bent Sbox is also APN.*

This result comes from the fact that, if the minimum distance of Γ_S is maximal (i.e., equal to $2^{n-1} - 2^{\frac{n-1}{2}}$), then the minimum distance of $\Gamma_S^\perp$ is also maximal (i.e., equal to 6). It is worth noticing that the converse does not hold in general, except for Sboxes of degree 2 [CCZ98].

Another approach used in [CV95] and revisited in [BN13] provides a relation between the differential spectrum and the squared Walsh coefficients. Indeed, the squared Walsh coefficients of an Sbox correspond to the Walsh transform of the entries in its difference table.

Theorem 2.14. *[CV95, BN13] Let S be a function from $\mathbb{F}_2^n$ into $\mathbb{F}_2^n$. Let $\delta(a, b)$ denote the entries in the difference table of S. Then, for any λ and μ in $\mathbb{F}_2^n$, we have*

$$\mathcal{E}^2(S_\mu + \varphi_\lambda) = \sum_{a \in \mathbb{F}_2^n} \sum_{b \in \mathbb{F}_2^n} (-1)^{a \cdot \lambda + b \cdot \mu} \delta(a, b) \ .$$

Conversely, for any a and b in $\mathbb{F}_2^n$,

$$\delta(a, b) = 2^{-2n} \sum_{\lambda \in \mathbb{F}_2^n} \sum_{\mu \in \mathbb{F}_2^n} (-1)^{a \cdot \lambda + b \cdot \mu} \mathcal{E}^2(S_\mu + \varphi_\lambda) \ .$$

Proof. We prove the second formula only, since the other one is directly derived by inverting the Walsh transform. Let a and b in $\mathbb{F}_2^n$. Then

$$\begin{aligned}
\mathcal{A} &= 2^{-2n} \sum_{\lambda\in\mathbb{F}_2^n} \sum_{\mu\in\mathbb{F}_2^n} (-1)^{a\cdot\lambda+b\cdot\mu} \mathcal{E}^2(S_\mu + \varphi_\lambda) \\
&= 2^{-2n} \sum_{\lambda\in\mathbb{F}_2^n} \sum_{\mu\in\mathbb{F}_2^n} (-1)^{a\cdot\lambda+b\cdot\mu} \left(\sum_{x\in\mathbb{F}_2^n} (-1)^{\mu\cdot S(x)+\lambda\cdot x} \right) \left(\sum_{y\in\mathbb{F}_2^n} (-1)^{\mu\cdot S(y)+\lambda\cdot y} \right) \\
&= 2^{-2n} \sum_{x\in\mathbb{F}_2^n} \sum_{y\in\mathbb{F}_2^n} \left(\sum_{\lambda\in\mathbb{F}_2^n} (-1)^{\lambda\cdot(x+y+a)} \right) \left(\sum_{\mu\in\mathbb{F}_2^n} (-1)^{\mu\cdot(S(x)+S(y)+b)} \right) \\
&= \#\{(x,y) \in \mathbb{F}_2^n \times \mathbb{F}_2^n : x+y=a \text{ and } S(x)+S(y)=b\} \\
&= \delta(a,b) \ .
\end{aligned}$$

◇

It is worth noticing that, while a function can be entirely recovered from the knowledge of its Walsh transform (or of its linear approximation table), this is not the case if only the difference table is known. Indeed, the difference table equivalently provides the magnitude of the Walsh coefficients, but does not give any information on the signs.

2.6 Sboxes with good cryptographic properties

We now give some examples of Sboxes with low linearity and differential uniformity. For small values of n, such Sboxes are classified. As previously explained for Boolean functions, this classification takes advantage of any equivalence relation under which the main cryptographic properties are invariant. In the case of Sboxes, the relevant relation is the *affine equivalence* in the following sense

2.6.1 Affine equivalence

Definition 2.15. *Two n-bit Sboxes S_1 and S_2 are said to be* affine equivalent *if there exist two affine permutations of $\mathbf{F}_2^n$, A_1 and A_2, such that*

$$S_2 = A_2 \circ S_1 \circ A_1$$

Proposition 2.16. *Let S_1 and S_2 be two affine equivalent Sboxes, with*

$$S_2 = A_2 \circ S_1 \circ A_1 \ .$$

Then, their difference tables and their linear-approximation tables satisfy

$$\begin{aligned}
\delta_{S_2}(a,b) &= \delta_{S_1}(L_1(a), L_2^{-1}(b)) \\
\mathcal{E}_{S_2}(a,b) &= \mathcal{E}_{S_1}((L_1^{-1})^T(a), L_2^T(b))
\end{aligned}$$

where L_1 and L_2 are the linear parts of A_1 and A_2.

Proof. Let $A_i(x) = L_i(x) + c_i$. We first consider the differential spectrum. An element $x \in \mathbb{F}_2^n$ satisfies $S_2(x+a) + S_2(x) = b$ if and only if

$$L_2\Big[S_1\big(L_1(x) + L_1(a) + c_1\big) + S_1\big(L_1(x) + c_1\big)\Big] = b\ ,$$

or equivalently

$$S_1\big(y + L_1(a)\big) + S_1(y) = L_2^{-1}(b)\ .$$

For the linear-approximation table, we use that

$$\begin{aligned} b \cdot S_2(x) + a \cdot x &= b \cdot c_2 + b \cdot L_2(S_1(y)) + a \cdot A_1^{-1}(y) \\ &= b \cdot c_2 + L_2^T(b) \cdot S_1(y) + a \cdot L_1^{-1}(c_1) + (L_1^{-1})^T(a) \cdot y\ . \end{aligned}$$

◇

2.6.2 Odd number of variables

When n is odd, almost bent Sboxes offer the best cryptographic properties since they have both the lowest possible linearity and differential uniformity. The list of all known *almost bent power permutations* of n variables, n odd, is given in Table 2.2.

	exponent s	
Quadratic exponents	$2^i + 1$ with $\gcd(i, n) = 1$, $1 \le i \le t$	[Gol68, Nyb93]
Kasami exponents	$2^{2i} - 2^i + 1$ with $\gcd(i, n) = 1$ $2 \le i \le t$	[Kas71]
Welch exponent	$2^t + 3$	[Dob99b, CCD00]
Niho exponent	$2^t + 2^{\frac{t}{2}} - 1$ is t is even $2^t + 2^{\frac{3t+1}{2}} - 1$ if t is odd	[Dob99a, HX01]

Table 2.2: Known almost bent power permutations $S : x \mapsto x^s$ over $\mathbb{F}_{2^n}$ with $n = 2t + 1$ (up to inversion).

A few AB permutations have been found which are not equivalent to a power permutation. For instance, when n is odd, divisible by 3 and not by 9.

$$S(x) = x^{2^i+1} + ux^{2^{j\frac{n}{3}}+2^{(3-j)\frac{n}{3}+i}} \text{ with } \gcd(i, n) = 1 \text{ and } j = i\frac{n}{3} \bmod 3$$

is an AB permutation [BCL08].

2.6.3 4-bit permutations

When n is even, almost bent functions do not exist and finding optimal functions is then much more difficult. For $n = 4$, 4-bit permutations have been classified up to affine equivalence

in [LP07] and in [De 07]. The smallest differential uniformity and linearity which can be achieved for a 4-bit permutation S are

$$\delta(S) \geq 4 \text{ and } \mathcal{L}(S) \geq 8 \, .$$

Up to affine equivalence, there are 16 classes of optimal permutations [LP07]. All of them have degree 3 but half of them have at least one component of degree 2 which may be an unsuitable property. Then, there are eight equivalence classes having all their non-trivial components of degree 3. A representative in each class is given in Table 2.3.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G_0	0	1	2	13	4	7	15	6	8	11	12	9	3	14	10	5
G_1	0	1	2	13	4	7	15	6	8	11	14	3	5	9	10	12
G_2	0	1	2	13	4	7	15	6	8	11	14	3	10	12	5	9
G_3	0	1	2	13	4	7	15	6	8	12	5	3	10	14	11	9
G_4	0	1	2	13	4	7	15	6	8	12	9	11	10	14	5	3
G_5	0	1	2	13	4	7	15	6	8	12	11	9	10	14	3	5
G_6	0	1	2	13	4	7	15	6	8	12	11	9	10	14	5	3
G_7	0	1	2	13	4	7	15	6	8	12	14	11	10	9	3	5
G_8	0	1	2	13	4	7	15	6	8	14	9	5	10	11	3	12
G_9	0	1	2	13	4	7	15	6	8	14	11	3	5	9	10	12
G_{10}	0	1	2	13	4	7	15	6	8	14	11	5	10	9	3	12
G_{11}	0	1	2	13	4	7	15	6	8	14	11	10	5	9	12	3
G_{12}	0	1	2	13	4	7	15	6	8	14	11	10	9	3	12	5
G_{13}	0	1	2	13	4	7	15	6	8	14	12	9	5	11	10	3
G_{14}	0	1	2	13	4	7	15	6	8	14	12	11	3	9	5	10
G_{15}	0	1	2	13	4	7	15	6	8	14	12	11	9	3	10	5

Table 2.3: The 8 classes of optimal 4-bit permutations [LP07].

2.6.4 6-bit permutations

It has been conjectured for a long time that APN permutations of an even number of variables did not exist. This has been disproved by Dillon in 2009 who has exhibited an APN permutation of 6 variables [BDMW10]. But this is (up to composition by affine transformations) the only known example of APN permutation of an even number of variables. This permutation is defined by

$$\begin{aligned} S \;=\; \{&0, 54, 48, 13, 15, 18, 53, 35, 25, 63, 45, 52, 3, 20, 41, 33, 59, 36, 2, 34, 10, 8, 57, 37, 60, 19, 42, 14, 50, \\ &26, 58, 24, 39, 27, 21, 17, 16, 29, 1, 62, 47, 40, 51, 56, 7, 43, 44, 38, 31, 11, 4, 28, 61, 46, 5, 49, 9, 6, 23, \\ &32, 30, 12, 55, 22\} \, . \end{aligned}$$

It satisfies

$$\delta(S) = 2 \, , \; \deg S = 4 \text{ and } \mathcal{L}(S) = 16 \, .$$

The corresponding univariate polynomial over $\mathbf{F}_{2^6}$ contains 53 nonzero monomials (out of the 56 possible monomials of degree at most 4), see e.g. [BN15, Page 131].

2.6.5 Sboxes of an even number of variables

For a permutation depending on an even number of variables n, $n \geq 8$, the best known differential uniformity is $\delta(S) = 4$. Then, we usually search for permutations S with

$$\delta(S) = 4 \text{ and } \mathcal{L}(S) = 2^{\frac{n+2}{2}} ,$$

even if we do not know whether these two values are optimal or not. Table 2.4 provides the list of all known power permutations over $\mathbb{F}_{2^n}$, n even, which achieve these values.

$2^i + 1$, $\gcd(i, n) = 2$	$n \equiv 2 \bmod 4$	[Gol68]
$2^{2i} - 2^i + 1$, $\gcd(i, n) = 2$	$n \equiv 2 \bmod 4$	[Kas71]
$2^n - 2$		[LW90]

Table 2.4: Known power permutations $S : x \mapsto x^s$ over $\mathbb{F}_{2^n}$ with $n = 2t$ (up to inversion) with the best known linearity and differential uniformity.

It is worth noticing that, when n is divisible by 4, the inversion is the only known permutation of n variables (up to composition by affine transformations) which reaches these two values.

Bibliography

[ADMS09] Jean-Philippe Aumasson, Itai Dinur, Willi Meier, and Adi Shamir. Cube testers and key recovery attacks on reduced-round MD6 and trivium. In *Fast Software Encryption - FSE 2009*, volume 5665 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2009.

[BCD11] Christina Boura, Anne Canteaut, and Christophe De Cannière. Higher-order differential properties of Keccak and Luffa. In *Fast Software Encryption - FSE 2011*, volume 6733 of *Lecture Notes in Computer Science*, pages 252–269. Springer, 2011.

[BCL08] Lilya Budaghyan, Claude Carlet, and Gregor Leander. Two classes of quadratic APN binomials inequivalent to power functions. *IEEE Transactions on Information Theory*, 54(9):4218–4229, 2008.

[BDMW10] K.A. Browning, J.F. Dillon, M.T. McQuistan, and A.J. Wolfe. An APN permutation in dimension six. In *Finite Fields: Theory and Applications*, volume 518 of *Contemporary Mathematics*, pages 33–42. AMS, 2010.

[Bla83] Richard E Blahut. *Theory and practice of error control codes*. Addison-Wesley, 1983.

[BN13] Céline Blondeau and Kaisa Nyberg. New links between differential and linear cryptanalysis. In *Advances in Cryptology - EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 388–404. Springer, 2013.

[BN15] Céline Blondeau and Kaisa Nyberg. Perfect nonlinear functions and cryptography. *Finite Fields and Their Applications*, 32:120–147, 2015.

[BS91] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.

[BT93] A.E. Brouwer and L.M.G.M. Tolhuizen. A sharpening of the Johnsson bound for binary linear codes and the nonexistence of linear codes with Preparata parameters. *Designs, Codes and Cryptography*, 3(2):95–98, 1993.

[BW72] Elwyn R. Berlekamp and Lloyd R. Welch. Weight distributions of the cosets of the (32, 6) Reed-Muller code. *IEEE Transactions on Information Theory*, 18(1):203–207, 1972.

[CAA+08] Carlos Cid, Martin Albrecht, Daniel Augot, Anne Canteaut, and Ralf-Philipp Weinmann. D.STVL.7 - algebraic cryptanalysis of symmetric primitives. Report

of the ECRYPT European Network of Excellence, July 2008. https://www.rocq.inria.fr/secret/Anne.Canteaut/Publications/dstvl7.pdf.

[Can02] Anne Canteaut. On the correlations between a combining function and functions of fewer variables. In *IEEE Information Theory Workshop - ITW 2002*, pages 78–81, Bangalore, Inde, October 2002. IEEE Press.

[CCD00] Anne Canteaut, Pascale Charpin, and Hans Dobbertin. Binary m-sequences with three-valued crosscorrelation: A proof of welch's conjecture. *IEEE Transactions on Information Theory*, 46(1):4–9, 2000.

[CCZ98] Claude Carlet, Pascale Charpin, and Victor Zinoviev. Codes, bent functions and permutations suitable for DES-like cryptosystems. *Designs, Codes and Cryptography*, 15(2):125–156, 1998.

[CHLL97] Gérard D. Cohen, Iiro S. Honkala, Simon Litsyn, and Antoine Lobstein. *Covering codes*. North-Holland, 1997.

[Cid04] Carlos Cid. Some Algebraic Aspects of the Advanced Encryption Standard. In *Advanced Encryption Standard - AES 2004*, volume 3373 of *Lecture Notes in Computer Science*, pages 58–66. Springer, 2004.

[CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Springer-Verlag, 2000.

[CP02] Nicolas Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In *Advances in Cryptology - ASIACRYPT'02*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer-Verlag, 2002.

[CT00] Anne Canteaut and Mickaël Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5. In *Advances in Cryptology - EUROCRYPT'2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 573–588. Springer-Verlag, 2000.

[CV95] Florent Chabaud and Serge Vaudenay. Links between differential and linear cryptanalysis. In *Advances in Cryptology - EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 356–365. Springer-Verlag, 1995.

[CW90] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic programming. *Journal of Symbolic Computation*, (9):251–280, 1990.

[De 07] Christophe De Cannière. *Analysis and Design of Symmetric Encryption Algorithms*. PhD thesis, KU Leuven, 2007.

[Del73] P. Delsarte. Four fundamental parameters of a code and their combinatorial signifiance. *Information and Control*, 23(5):407–438, décembre 1973.

[Dob94] Hans Dobbertin. Construction of bent functions and balanced Boolean functions with high nonlinearity. In *Fast Software Encryption - FSE'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 61–74. Springer-Verlag, 1994.

[Dob99a] Hans Dobbertin. Almost perfect nonlinear power functions on $GF(2^n)$: the Niho case. *Information and Computation*, 151(1-2):57–72, 1999.

[Dob99b] Hans Dobbertin. Almost perfect nonlinear power functions on $GF(2^n)$: the Welch case. *IEEE Transactions on Information Theory*, 45(4):1271–1275, 1999.

[DS09] Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In *Advances in Cryptology - EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299. Springer, 2009.

[DZ84] Stefan M. Dodunekov and Victor Zinoviev. A note on Preparata codes. In *Proceedings of the 6th Intern. Symp. on Information Theory, Moscow-Tashkent Part 2*, pages 78–80, 1984.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, 1999.

[Fau02] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*. ACM, 2002.

[GC91] Henri Gilbert and Guy Chassé. A Statistical Attack of the FEAL-8 Cryptosystem. In *Advances in Cryptology - CRYPTO'90*, volume 537 of *Lecture Notes in Computer Science*, pages 22–33. Springer-Verlag, 1991.

[GG04] Solomon W. Golomb and Guang Gong. *Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar*. Cambridge University Press, 2004.

[Gol68] Robert Gold. Maximal recursive sequences with 3-valued recursive crosscorrelation functions. *IEEE Transactions on Information Theory*, 14:154–156, 1968.

[Hou93] Xiang-Dong Hou. Further results on the covering radii of the Reed-Muller codes. *Designs, Codes and Cryptography*, 3:167–177, 1993.

[Hou96a] Xiang-Dong Hou. Covering radius of the Reed-Muller code $R(1, 7)$ - a simpler proof. *Journal of Combinatorial Theory, Series A*, (74):337–341, 1996.

[Hou96b] Xiang-Dong Hou. On the covering radius of $R(1, m)$ in $R(3, m)$. *IEEE Transactions on Information Theory*, 42(3):1035–1037, 1996.

[HX01] H.D.L. Hollmann and Q. Xiang. A proof of the Welch and Niho conjectures on crosscorrelations of binary m-sequences. *Finite Fields and their Applications*, 7(2):253–286, 2001.

[JK97] Thomas Jakobsen and Lars R. Knudsen. The interpolation attack on block ciphers. In *Fast Software Encryption - FSE'97*, volume 1267 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[Jou09] Antoine Joux. *Algorithmic Cryptanalysis*. Chapman & Hall/CRC, 2009.

[Kas71] Tadao Kasami. The weight enumerators for several classes of subcodes of the second order binary Reed-Muller codes. *Information and Control*, 18:369–394, 1971.

[KMY07] Selçuk Kavut, Subhamoy Maitra, and Melek D. Yücel. Search for Boolean Functions With Excellent Profiles in the Rotation Symmetric Class. *IEEE Transactions on Information Theory*, 53(5):1743–1751, 2007.

[Knu95] Lars R. Knudsen. Truncated and higher order differentials. In *Fast Software Encryption - FSE'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 1995.

[KR11] Lars R. Knudsen and Matthew Robshaw. *The Block Cipher Companion*. Information Security and Cryptography. Springer, 2011.

[LP07] Gregor Leander and Axel Poschmann. On the Classification of 4 Bit S-Boxes. In *Arithmetic of Finite Fields - WAIFI 2007*, volume 4547 of *Lecture Notes in Computer Science*, pages 159–176. Springer, 2007.

[LW90] G. Lachaud and J. Wolfmann. The weights of the orthogonal of the extended quadratic binary Goppa codes. *IEEE Transactions on Information Theory*, 36(3):686–692, 1990.

[Mat94] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[Mat95] Mitsuru Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In *Advances in Cryptology - CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.

[McE72] Robert J. McEliece. Weight congruence for p-ary cyclic codes. *Discrete Mathematics*, 3:177–192, 1972.

[MM94] Oscar Moreno and Carlos J. Moreno. The MacWilliams-Sloane conjecture on the tightness of the Carlitz-Uchiyama bound and the weights of duals of BCH codes. *IEEE Transactions on Information Theory*, 40(6):1894–1907, 1994.

[Moe12] Möbius inversion. Encyclopedia of Mathematics, 2012. `http://www.encyclopediaofmath.org/index.php?title=M%C3%B6bius_inversion&oldid=23416`.

[MS77] F. Jessie MacWilliams and Neil J.A. Sloane. *The theory of error-correcting codes*. North-Holland, 1977.

[Mul54] David E. Muller. Application of Boolean algebra to switching circuit design and to error detection. *IEEE Transactions on Computers*, 3:6–12, 1954.

[Myk80] Johannes Mykkeltveit. The covering radius of the (128,8) Reed-Muller code is 56. *IEEE Transactions on Information Theory*, IT-26(3):359–362, 1980.

[NK93] Kaisa Nyberg and Lars R. Knudsen. Provable security against differential cryptanalysis. In *Advances in Cryptology - CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 566–574. Springer-Verlag, 1993.

[Nyb91] Kaisa Nyberg. Perfect nonlinear S-boxes. In *Advances in Cryptology - EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 378–385. Springer-Verlag, 1991.

[Nyb93] Kaisa Nyberg. Differentially uniform mappings for cryptography. In *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 55–64. Springer-Verlag, 1993.

[Plo60] Morris Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960.

[PW83] Nick J. Patterson and Douglas H. Wiedemann. The covering radius of the $[2^{15}, 16]$ Reed-Muller code is at least 16276. *IEEE Transactions on Information Theory*, IT-36(2):443, 1983.

[Ree54] Irving S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *IEEE Transactions on Information Theory*, 4:38–49, 1954.

[Rot64] Gian-Carlo Rota. On the foundations of combinatorial theory I. Theory of Möbius Functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 2(4):340–368, 1964.

[SB70] Neil J. A. Sloane and Elwyn R. Berlekamp. Weight enumerator for second-order Reed-Muller codes. *IEEE Transactions on Information Theory*, 16(6):745–751, 1970.

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, 1949.

[Sie84] Thomas Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, IT-30(5):776–780, 1984.

[SM66] Gustave Solomon and Robert J. McEliece. Weights of cyclic codes. *Journal of Combinatorial Theory*, 1(4), 1966.

[SM95] Douglas R. Stinson and James L. Massey. An infinite class of counterexamples to a conjecture concerning nonlinear resilient functions. *Journal of Cryptology*, 8(3):167–173, 1995.

[Ste04] Allan Steel. Allan Steel's Gröbner basis timings page, 2004. `http://magma.maths.usyd.edu.au/users/allan/gb/`.

[TCG91] Anne Tardy-Corfdir and Henri Gilbert. A known plaintext attack of FEAL-4 and FEAL-6. In *Advances in Cryptology - CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 172–182. Springer-Verlag, 1991.

[Weg87] Ingo Wegener. *The complexity of Boolean functions*. Wiley-Teubner, 1987.

[XM88] Guozheng Xiao and James L. Massey. A spectral characterization of correlation-immune combining functions. *IEEE Transactions on Information Theory*, IT-34(3):569–571, 1988.

[Zha00] Muxiang Zhang. Maximum correlation analysis of nonlinear combining functions in stream ciphers. *Journal of Cryptology*, 13(3):301–313, 2000.