Yet another algorithm to compute the nonlinearity of a Boolean function

Dott. Emanuele Bellini

University of Trento, CryptoLab.
Telsy Elettronica S.p.A. (Turin)

YACC, June 2014

Index

Hamming distance

Let $\mathbb{F}$ denote the binary field $\mathbb{F}_2$.
The set $\mathbb{F}^n$ is the set of all binary vectors of length n.

Definition

Let $v \in \mathbb{F}^n$.
The *Hamming weight* $\mathrm{w}(v)$ of the vector v is the number of its nonzero coordinates.
For any two vectors $v_1, v_2 \in \mathbb{F}^n$, the *Hamming distance* between v_1 and v_2, denoted by $\mathrm{d}(v_1, v_2)$, is the number of coordinates in which the two vectors differ.

Boolean functions

Definition

A *Boolean function* (B.f.) is any function $f : \mathbb{F}^n \to \mathbb{F}$.
The set of all B.f. 's from $\mathbb{F}^n$ to $\mathbb{F}$ will be denoted by $\mathcal{B}_n$.

B.f. can be represented in a unique way in many different forms:

1. Algebraic normal form
2. Truth table (evaluation vector)
3. Numerical normal form
4. Walsh spectrum
5. ...

Evaluation vector

We assume implicitly to have ordered $\mathbb{F}^n$, so that

$$\mathbb{F}^n = \{\mathrm{p}_1, \ldots, \mathrm{p}_{2^n}\}.$$

Definition

We consider the evaluation map from $\mathcal{B}_n$ to $\mathbb{F}^{2^n}$, associating to each B.f. f the vector $\underline{f} = (f(\mathrm{p}_1) \ldots, f(\mathrm{p}_{2^n}))$, which is called the *evaluation vector* of f.

The evaluation vector of f uniquely identifies f.

Algebraic normal form

Proposition

A B.f. $f \in \mathcal{B}_n$ can be expressed in a unique way as a polynomial in $\mathbb{F}[X] = \mathbb{F}[x_1, \ldots, x_n]$, as

$$f = \sum_{v \in \mathbb{F}^n} b_v X^v ,$$

where $X^v = x_1^{v_1} \cdots x_n^{v_n}$.

This representation is called the *Algebraic Normal Form* (ANF).

Numerical normal form

In 1999, Carlet introduced a useful representation of B.f. 's for characterizing several cryptographic criteria.

B.f. 's can be represented as elements of $\mathbb{K}[X]/\langle X^2 - X\rangle$, where $\langle X^2 - X\rangle$ is the ideal generated by the polynomials $x_1^2 - x_1, \dots, x_n^2 - x_n$, and $\mathbb{K}$ is $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, or $\mathbb{C}$.

Numerical normal form

Definition

Let f be a function on $\mathbb{F}^n$ taking values in a field $\mathbb{K}$.
We call the *numerical normal form* (NNF) of f the following expression of f as a polynomial:

$$f(x_1, \ldots, x_n) = \sum_{u \in \mathbb{F}^n} \lambda_u X^u ,$$

with $\lambda_u \in \mathbb{K}$ and $u = (u_1, \ldots, u_n)$.

Once $\mathbb{K}$ is fixed, it can be proved that any B.f. f admits a unique NNF.

Nonlinearity of a Boolean function

Let $f, g \in \mathcal{B}_n$. The distance $\mathrm{d}(f,g)$ between f and g is the number of $v \in \mathbb{F}^n$ such that $f(v) \neq g(v)$. It is obvious that $\mathrm{d}(f,g) = \mathrm{d}(\underline{f}, \underline{g}) = \mathrm{w}(\underline{f} + \underline{g})$.

Definition

Let $f \in \mathcal{B}_n$. The *nonlinearity* of f is the minimum of the distances between f and any affine function, i.e. $\mathrm{N}(f) = \min_{\alpha \in \mathcal{A}_n} \mathrm{d}(f, \alpha)$.

Nonlinearity of a Boolean function

Proposition

The maximum nonlinearity for a B.f. f is bounded by
$\max\{\mathrm{N}(f) \mid f \in \mathcal{B}_n\} \leq 2^{n-1} - 2^{\frac{n}{2}-1}$.

Walsh spectrum

Definition

The *Walsh transform* of a B.f. $f \in \mathcal{B}_n$ is a function $\hat{F} : \mathbb{F}^n \to \mathbb{Z}$ such that

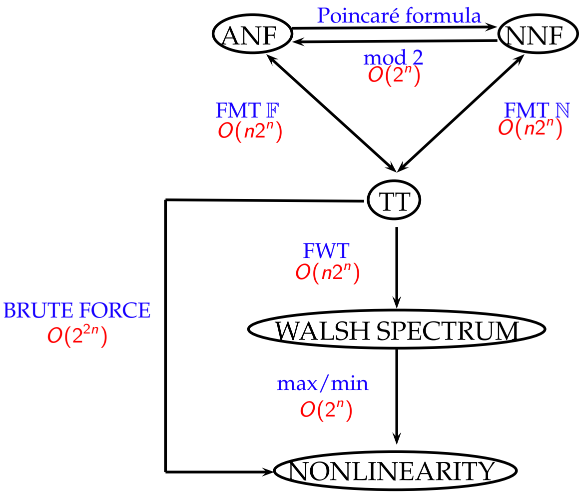
$$\hat{F}(x) = \sum_{y \in \mathbb{F}^n} (-1)^{x \cdot y + f(y)},$$

where $x \cdot y$ is the scalar product of x and y.

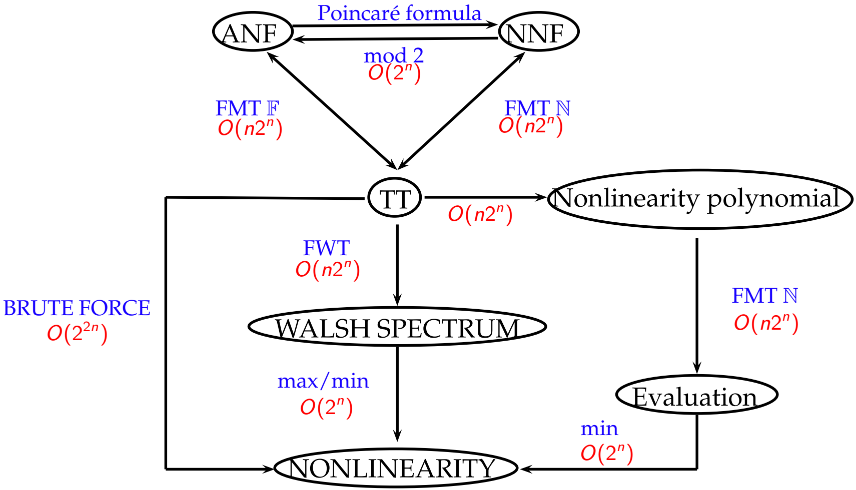
The set of integers $\{\hat{F}(v) \mid v \in \mathbb{F}^n\}$ is called the *Walsh spectrum* of the B.f. f. It holds that

$$\mathrm{N}(f) = \min_{v \in \mathbb{F}^n} \{2^{n-1} - \frac{1}{2}\hat{F}(v)\} = 2^{n-1} - \frac{1}{2}\max_{v \in \mathbb{F}^n} \{\hat{F}(v)\}.$$

Cost of changing representation



Cost of changing representation



Generic affine polynomial

Let A be the variable set $A = \{a_i\}_{0 \leq i \leq n}$. We denote by $\mathfrak{g}_n \in \mathbb{F}[A, X]$ the following polynomial:

$$\mathfrak{g}_n = a_0 + \sum_{i=1}^{n} a_i x_i \ .$$

Determining the nonlinearity of $f \in \mathcal{B}_n$ is the same as finding the minimum weight of the vectors in the set $\{\underline{f} + \underline{g} \mid g \in \mathcal{A}_n\} \subset \mathbb{F}^{2^n}$. We can consider the evaluation vector of the polynomial $\mathfrak{g}_n$ as follows:

$$\underline{\mathfrak{g}_\mathfrak{n}} = (\mathfrak{g}_n(A, \mathrm{p}_1), \ldots, \mathfrak{g}_n(A, \mathrm{p}_{2^n})) \in (\mathbb{F}[A])^{2^n} \ .$$

The nonlinearity polynomial

For each $0 \leq i \leq 2^n$, we define the following Boolean affine polynomials:

$$f_i^{(\mathbb{F})}(A) = \mathfrak{g}_n(A, \mathrm{p}_i) + f(\mathrm{p}_i)\,.$$

We also define

$$f_i^{(\mathbb{Z})}(A) = \mathrm{NNF}(f_i^{(\mathbb{F})}(A))\,.$$

Definition

We call $\mathfrak{n}_f(A) = f_1^{(\mathbb{Z})}(A) + \cdots + f_n^{(\mathbb{Z})}(A) \in \mathbb{Z}[A]$ the **nonlinearity polynomial** (NLP) of the B.f. f.

Notice that the integer evaluation vector $\underline{\mathfrak{n}_f}$ represents all the distances of f from all possible affine functions in n variables.

Computing the nonlinearity

Thus, to compute the nonlinearity of f we have to find the minimum nonnegative integer t in the set of the evaluations of $\mathfrak{n}_f$, that is, in $\{\mathfrak{n}_f(\bar{a}) \mid \bar{a} \in \{0,1\}^{n+1} \subset \mathbb{Z}^{n+1}\}$.

The nonlinearity ideal

Definition

For any $t \in \mathbb{N}$ we define the ideal $\mathcal{N}_f^t \subseteq \mathbb{Q}[A]$ as follows:

$$\mathcal{N}_f^t = \langle E[A] \bigcup \{ f_1^{(\mathbb{Z})} + \cdots + f_{2^n}^{(\mathbb{Z})} - t \} \rangle = \langle E[A] \bigcup \{ \mathfrak{n}_f - t \} \rangle \quad (1)$$

Theorem

The variety of the ideal $\mathcal{N}_f^t$ is non-empty if and only if the Boolean function f has distance t from an affine function. In particular, $\mathrm{N}(f) = t$, where t is the minimum positive integer such that $\mathcal{V}(\mathcal{N}_f^t) \neq \emptyset$.

A first algorithm using Gröbner basis

Input: f
Output: nonlinearity of f

1: Compute $\mathfrak{n}_f$
2: $j \leftarrow 1$
3: **while** $\mathcal{V}(\mathcal{N}_f^j) = \emptyset$ **do**
4: $\quad j \leftarrow j + 1$
5: **end while**
6: **return** j

Example

Consider the case $n = 2$, $f(x_1, x_2) = x_1x_2 + 1$. We have that $\underline{f} = (1, 1, 1, 0)$ and $\underline{\mathfrak{g}_n} = (a_0, a_0 + a_1, a_0 + a_2, a_0 + a_1 + a_2)$.
Let us compute all $f_i^{(\mathbb{F})} = (\underline{\mathfrak{g}_n} + \underline{f})_i$ and $f_i^{(\mathbb{Z})}$,for $i = 1, \ldots, 2^2$:

$$
\begin{aligned}
f_1^{(\mathbb{F})} &= a_0 + 1 & \rightarrow f_1^{(\mathbb{Z})} &= -a_0 + 1 \\
f_2^{(\mathbb{F})} &= a_0 + a_1 + 1 & \rightarrow f_2^{(\mathbb{Z})} &= 2a_0a_1 - a_0 - a_1 + 1 \\
f_3^{(\mathbb{F})} &= a_0 + a_2 + 1 & \rightarrow f_3^{(\mathbb{Z})} &= 2a_0a_2 - a_0 - a_2 + 1 \\
f_4^{(\mathbb{F})} &= a_0 + a_1 + a_2 & \rightarrow f_4^{(\mathbb{Z})} &= 4a_0a_1a_2 - 2a_0a_1 - 2a_0a_2 + a_0 \\
& & &\quad - 2a_1a_2 + a_1 + a_2
\end{aligned}
$$

Example

Then $\mathfrak{n}_f = f_1^{(\mathbb{Z})} + f_2^{(\mathbb{Z})} + f_3^{(\mathbb{Z})} + f_4^{(\mathbb{Z})} = 4a_0a_1a_2 - 2a_0 - 2a_1a_2 + 3$ and since

$$\underline{\mathfrak{n}_f} = (3, 1, 3, 1, 3, 1, 1, 3)$$

then the nonlinearity of f is 1.
Notice that the vector $\underline{\mathfrak{n}_f}$ represents all the distances of f from all possible affine functions in 2 variables, that is, from $0, 1, x_1, x_1 + 1, x_2, x_2 + 1, x_1 + x_2, x_1 + x_2 + 1$.

Coefficients of the NLP

Theorem

Let $v = (v_0, v_1, \ldots, v_n) \in \mathbb{F}^{n+1}$, $\tilde{v} = (v_1, \ldots, v_n) \in \mathbb{F}^n$, $A^v = a_0^{v_0} \cdots a_n^{v_n} \in \mathbb{F}[A]$ *and* $c_v \in \mathbb{Z}$ *be such that* $\mathfrak{n}_f = \sum_{v \in \mathbb{F}^{n+1}} c_v A^v$. *Then the coefficients of* $\mathfrak{n}_f$ *can be expressed as:*

$$c_v = \sum_{u \in \mathbb{F}^n} f(u) = \mathrm{w}(\underline{f}) \quad \text{if } v = 0 \qquad (2)$$

$$c_v = (-2)^{\mathrm{w}(v)} \sum_{\substack{u \in \mathbb{F}^n \\ \tilde{v} \preceq u}} \left[f(u) - \frac{1}{2} \right] \quad \text{if } v \neq 0 \qquad (3)$$

Algorithm to compute the nonlinearity polynomial

Input: The evaluation vector $\underline{f}$ of a B.f. $f(x_1, \ldots, x_n)$
Output: the vector $c = (c_1, \ldots, c_{2n+1})$ of the coefficients of $\mathfrak{n}_f$

Calculation of the coefficients of the monomials *not* containing a_0

$(c_1, \ldots, c_{2n}) = \underline{f}$
for $i = 0, \ldots, n-1$ **do**
  $b \leftarrow 0$
  repeat
    for $x = b, \ldots, b + 2^i - 1$ **do**
      $c_{x+1} \leftarrow c_{x+1} + c_{x+2^i+1}$
      if $x = b$ **then**
        $c_{x+2^i+1} \leftarrow 2^i - 2c_{x+2^i+1}$
      else
        $c_{x+2^i+1} \leftarrow -2c_{x+2^i+1}$
      end if
    end for
    $b \leftarrow b + 2^{i+1}$
  until $b = 2^n$
end for

Calculation of the coefficients of the monomials containing a_0

$c_{1+2^n} \leftarrow 2^n - 2c_1$
for $i = 2, \ldots, 2^n$ **do**
  $c_{i+2^n} \leftarrow -2c_i$
end for
return c

A butterfly scheme to compute the coefficients of the NLP

(x_1, x_2, x_3)	$f(x_1, x_2, x_3)$	*Step* 1	*Step* 2	*Step* 3
000	e_1	$e_1 + e_2$	$e_1 + e_2 + e_3 + e_4$	$e_1 + e_2 + e_3 + e_4 + e_5 + e_6 + e_7 + e_8$
001	e_2	$1 - 2e_2$	$2 - 2e_2 - 2e4$	$4 - 2e_2 - 2e_4 - 2e_6 - 2e_8$
010	e_3	$e_3 + e_4$	$2 - 2e_3 - 2e4$	$4 - 2e_3 - 2e_4 - 2e_7 - 2e_8$
011	e_4	$1 - 2e_4$	$-2 + 4e_4$	$-4 + 4e_4 - 4e_8$
100	e_5	$e_5 + e_6$	$e_5 + e_6 + e_7 + e_8$	$4 - 2e_5 - 2e_6 - 2e_7 - 2e_8$
101	e_6	$1 - 2e_6$	$2 - 2e_6 - 2e8$	$-4 + 4e_6 - 4e_8$
110	e_7	$e_7 + e_8$	$2 - 2e_7 - 2e8$	$-4 + 4e_7 - 4e_8$
111	e_8	$1 - 2e_8$	$-2 + 4e_8$	$4 - 8e_8$

Complexity of computing the NLP coefficients

Theorem

Computing the coefficients of the nonlinearity polynomial requires $O(n2^n)$ integer sums and doublings, in particular circa $n2^{n-1}$ integer sums and circa $n2^{n-1}$ integer doublings, and the storage of $O(2^n)$ integers of size less than or equal to 2^n.

Complexity of computing the nonlinearity using the NLP

Theorem

Determining the coefficients of the polynomial $\mathfrak{n}_f$ from the truth table of f and then finding $\mathrm{N}(f) = \min\{\mathfrak{n}_f(\bar{a}) \mid \bar{a} \in \{0,1\}^{n+1}\}$ requires a total $O(n2^n)$ integer operations (sums and doublings).

Some experimental comparison

n	4-5	5-6	6-7	7-8	8-9	9-10	10-11
$\log_2 \left[\frac{(n+1)2^{n+1}}{n2^n}\right]$	1.22	1.17	1.14	1.12	1.11	1.09	1.09
FWT	0.90	0.98	1.01	1.22	0.95	1.25	1.07
NLP+FPE	1.02	1.09	1.13	1.07	1.17	1.07	1.11

Conclusion and future work

With a different approch we are able to compute the nonlinearity of a B.f. with the same complexity as classical methods.

- Is $O(n2^n)$ the complexity of the problem?
- How to compute the ANF or the evaluation vector of a B.f. from its nonlinearity polynomial?
- Are there similar methods to compute other properties of a B.f. (weight, resiliency, etc.)?
- The method can be extended to compute the minimum weight of any nonlinear binary code. Are there cases where the method is faster than brute force or than Brouwer-Zimmerman probabilistic algorithm for linear codes?

Grazie per l'attenzione!