Calculating Nonlinearity of Boolean Functions with Walsh-Hadamard Transform

Pedro Miguel Sosa
UCSB, Santa Barbara, CA - USA

April 23, 2016

Abstract

In this paper we will explain the meaning of nonlinearity in boolean functions, and how to calculate it using Walsh Transform. We will start by providing a brief background on Boolean Functions. We will explain the process by which we calculate nonlinearity and also provide an explanation of the Fast Walsh-Hadamard Transform, which will allow for quicker implentation of this calculation. To conclude, we will provide some motivation behind the importance of nonlinearity and give an overview of Bent Functions.

1 Boolean Functions

Overview Boolean functions are commonly used and quite important in the area of Cryptography. They can be mostly found in the construction of symmetric key algorithms (e.g Substitution Boxes, Linear Feedback Registers, Random Number Generator, among others).

Definition Mathematically, Boolean functions are defined with the form: $f_k : B_2^k \Rightarrow B_2$, such that $B_2 = \{0, 1\}$ and $k \in \mathbb{Z}$. The k value is referred to as *arity* of the function and it simply defines how many variables it takes as input. Furthermore, with boolean functions, the additive function is defined with XOR and the multiplicative function is defined with AND.

Affine Boolean Functions Affine boolean functions are the subset of boolean functions that can be described as:

$$f = a_k x_k \oplus a_{k-1} x_{k-1} \oplus ... \oplus a_1 x_1 + a_0$$

Balance One of the most important and basic properties that cryptographers seek when choosing to work with certain boolean functions is *Balance*. This will certainly be important when defining nonlinearity. Balanced functions are those which output **0**s and **1**s with same probability (50% each) for any given input.

Distance Between Functions Another important concept to understand nonlinearity is the distance between boolean functions. The distance between two functions f and g represents how many bits need to be inverted on the truth table of function f to match that of function g.

$$example: \quad f(x) = 0\,1\,\mathbf{0}\,\mathbf{1}\,\mathbf{0}\,1\,1\,\mathbf{1}\,\mathbf{1}\,0\,1\,0$$
$$g(x) = 0\,1\,\mathbf{1}\,\mathbf{0}\,\mathbf{1}\,1\,1\,\mathbf{0}\,\mathbf{0}\,0\,1\,0$$

$$distance(f, g) = 5$$

2 Nonlinearity

Motivation The main reason we would want to define the nonlinearity of a given function is to understand how easy it would be to find a correlation between this function's input and its output. These correlations could then be used to attack any crypto-system that uses this function.

Definition Nonlinearity can be defined as the number of bits that we must invert on a given function f_k truth table to reach the closest k-arity affine function a_k.

2.1 Calculating Nonlinearity

Expected Distance Assume we have a function f_k that we want to test against an affine function a_k. when we compare the distance between each function, we expect it to be: $\frac{(2^k)}{2}$. In other words, we expect about half the bits to be different between the two functions.

Notice that while it might seem counter intuitive, having a lesser distance between f_k and a_k would simply indicate that f_k is actually close to the inverse of a_k (which can be defined as: $a_k^{-1} = a_k \oplus 1$).

$$f(x) = 1\mathbf{0000111} \qquad f(x) = 1\mathbf{0}0000\mathbf{11}$$
$$a(x) = \mathbf{0}0\mathbf{111}00 \qquad a^1(x) = 1\mathbf{1}0000\mathbf{1}1$$
$$dist(f(x), a(x)) = 6 \qquad dist(f(x), a^{-1}(x)) = 2$$

Thus a distance of $\frac{(2^k)}{2}$ is the best we can expect.

Unexpected Distance We will define the *unexpected distance* [4] as the amount by which the $distance(f, g)$ differs from our expected distance.

$$example: \quad f_3 = \mathbf{10}100\mathbf{11}$$
$$g_3 = \mathbf{01}101\mathbf{00}$$

In the example above the expected distance would be $\frac{2^3}{2} = 4$ and the unexpected distance would be $4 - 6 = -2$ which means that f had 2 more extra bits of difference that we weren't expecting.

Walsh-Hadamard Transform The Walsh-Hadamard Transform (WHT) is a generalized class of Fourier Transform, which will calculate the unexpected distances between a given function f_k and all k-arity affine functions.

This transform will take the f_k's truth table (TT), multiply it against the Hadamard matrix, and return the Walsh spectrum of the function (which in our case will represent the unexpected distances between f and all k-arity affine functions).

$$\big[f\text{'s TT}\big] \cdot \big[\text{Hadamard Matrix}\big] = \big[\text{Walsh Spect.}\big]$$

Building the Hadamard matrix The Hadamard matrix is defined as a nxn matrix whose entries are mutually orthogonal and can only take values of -1 or 1.

Hadamard matrices are defined recursively by the following sequence:

$$H_1 = \big[1\big]; H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix};$$

$$H_{2^k} = \begin{bmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{bmatrix} = H_2 \otimes H_{2^{k-1}},$$

If ones is to carefully examine the rows of these matrices, one will find that they represent the truth table for a particular affine function of size n. Thus, to solve a k-arity function one will need to use the Hadamard matrix of size $n = 2^k$.

Fast Walsh Hadamard The fast Hadamard Transform is a Divide & Conquer algorithm to compute the Walsh-Hadamard function with a computational complexity of O*(N log N)*

FTW procedure can be understood as follows, (where FTT refers to the f_k Truth Table):

```
procedure FWH(FTT)
    n = size(FTT)
    for r = 1 to (n/2-1):
        for i = 0 to (n-r):
            FTT[i] = FTT[i] + FTT[i+r]
            FTT[i+r] = FTT[i] - FTT[i+r]
end procedure
```

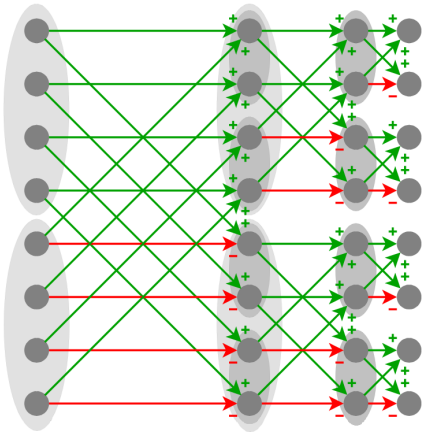

Figure 1: Fast Walsh Hadamard Transform [1]

[1]Image by: Timako

Calculating Nonlinearity Once the Walsh spectrum has been calculated we can measure nonlinearity by the formula:

$$N(f_k) = 2^{k-1} - \frac{1}{2}\max_{a \in \mathbb{F}_{2^k}} |W_f(a)|$$

Essentially, nonlinearity is the difference between our expected distance 2^{k-1} and the absolute value of the maximum unexpected distance we found between our f_k and some affine function a_k.

3 Example Calculation

Assume one wishes to test the non-linearity of a function $f_{k=3}$ whose truth table is:

$$f = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Thus, one would compute the 8x8 Hadamard Matrix to be:

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

One then proceeds to calculate the Walsh spectrum by multiplying $f_{k=3}$'s truth table with out Hadamard Matrix.

$$f \cdot H_8 = \begin{bmatrix} 4 & 2 & -2 & 0 & 0 & -2 & -2 & 0 \end{bmatrix}$$

Finally, one can see that the maximum unexpected distance is $|2|$. Notice that one disregards the first element (e.i. 4) of the Walsh Spectrum since this element actually refers to

the $weight(f)$. So the final nonlinearity calculation becomes:

$$N(f_{k=3}) = 2^{3-1} - |2| = \mathbf{2}$$

**Note:* one could use a truth table described with $\{True = 1, False = -1\}$. This would yield a spectrum where all values are 2·unexpected distance. [5]

4 Bent Functions

4.1 Definition

In Cryptography we are interested in finding the functions with highest possible nonlinearity. By definition, the nonlinearity of a k-ary function f ranges from 0 (meaning f is itself an affine function.) to $2^{k-1} - 2^{\frac{k}{2}-1}$. Functions that achieve this upper bound are considered **bent**.

4.2 Characteristics of Bent Functions

Propagation Criteria Bent functions will satisfy the Propagation Criteria (PC) of degree k. The propagation criteria is a higher-order formalization of the Avalanche effects. PC(k) refers to the behavior observed in boolean functions were for any given input, if n bits ($0 < n < k$) are flipped each of the output bits will change with a 50% probability.

Mathematically we can define this as:

$$f \text{ is } fPC(n) \text{ if } \forall\, a \text{ with } w_H(a) \leq n$$

$$D_a f(x) = f(x) \oplus f(x \oplus a) \text{is balanced}$$

Generation of Hadamard Matrix Given any bent function f_k we are able to generate the Hadamard Matrix. The process to do this is defined as:

$$H_k = [(-1)^{f(x \oplus y)}]_{x,y \in \mathbb{F}_2^k}$$

4.3 Characterization of Bent Functions

The most common characterization of bent functions is the mod m type, defined as: [6]

$$f : \mathbb{Z}_m^n \to \mathbb{Z}_m$$

$$\hat{f}(a) = \sum_{x \in \mathbb{Z}_m^n} e^{\frac{2\pi i}{m}(f(x) - a \cdot x)}$$

However, there is still no complete characterization of bent functions to date, partially because these are quite atypical among higher arity boolean functions. This is still considered an open problem within Cryptography.

References

[1] C. Carlet. *Encyclopedia of Cryptography and Security*, chapter Nonlinearity of Boolean Functions, pages 416–417. Springer US, Boston, MA, 2005.

[2] C. Carlet. Boolean functions for cryptography and error correcting codes, 2007.

[3] B. Fino and V. Algazi. Unified matrix treatment of the fast walsh-hadamard transform. *Computers, IEEE Transactions on*, C-25(11):1142–1146, Nov 1976.

[4] T. R. O'Downd. Discovery of bent functions using the fast walsh transform. Master's thesis, Naval Postgraduate School (U.S.), Monterey, California, 12 2010.

[5] T. Ritter. Measuring boolean function nonlinearity by walsh transform, 1998.

[6] O. Rothaus. On "bent" functions. *Journal of Combinatorial Theory*, 20(3):300–305, May 1976.