Cryptography – laboratory – exercise 1-2.
Quality assessment of an S-box. Nonlinearity and SAC .

Definitions:

We use square brackets to denote vectors like $[a_1, ..., a_n]$ and round brackets to denote functions like $f(x_1, ..., x_n)$.

Boolean Function
Let GF(2) = $<\Sigma, \oplus, \bullet>$ be two-element Galois field, where $\Sigma = \{0, 1\}$, $\oplus$ and $\bullet$ denotes the sum and multiplication mod 2, respectively. A function f: $\Sigma^n \to \Sigma$ is an n-argument Boolean function. Let $z = x_1 \cdot 2^{n-1} + x_2 \cdot 2^{n-2} + ... + x_n \cdot 2^0$ be the decimal representation of arguments $(x_1, x_2, ..., x_n)$ of the function f. Let us denote $f(x_1, x_2, ..., x_n)$ as y_z. Then $[y_0, y_1, \ldots, y_{2^n-1}]$ is called a truth table of the function f.

Linear and Nonlinear Boolean Functions
An n-argument Boolean function f is linear if it can be represented in the following form: $f(x_1, x_2, ..., x_n) = a_1x_1 \oplus a_2x_2 \oplus ... \oplus a_nx_n$. Let L_n be a set of all n-argument linear Boolean functions. Let $M_n = \{g: \Sigma^n \to \Sigma \,/\, g(x_1, x_2, ..., x_n) = 1 \oplus f(x_1, x_2, ..., x_n)$ and $f \in L_n\}$. A set $A_n = L_n \cup M_n$ is called a set of n-argument affine Boolean functions. A Boolean function f: $\Sigma^n \to \Sigma$ that is not affine is called a nonlinear Boolean function.

Balance
Let $N_0[y_0, y_1, \ldots, y_{2^n-1}]$ be a number of zeros (0's) in the truth table $[y_0, y_1, \ldots, y_{2^n-1}]$ of function f, and $N_1[y_0, y_1, \ldots, y_{2^n-1}]$ be number of ones (1's). A Boolean function is balanced if $N_0[y_0, y_1, \ldots, y_{2^n-1}] = N_1[y_0, y_1, \ldots, y_{2^n-1}]$.

Algebraic Normal Form
A Boolean function can also be represented as a maximum of 2^n coefficients of the Algebraic Normal Form. These coefficients provide a formula for the evaluation of the function for any given input $x = [x_1, x_2, ..., x_n]$:

$$f(\mathrm{x}) = a_0 \oplus \sum_{i=1}^{n} a_i x_i \oplus \sum_{1 \le i < j \le n} a_{ij} x_i x_j \oplus \ldots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n$$

where $\sum, \oplus$ denote the modulo 2 summations.

The order of nonlinearity of a Boolean function $f(x)$ is a maximum number of variables in a product term with non-zero coefficient a_J, where J is a subset of $\{1, 2, 3, ..., n\}$. In the case where J is an empty set the coefficient is denoted as a_0 and is called a zero-order coefficient. Coefficients of order 1 are $a_1, a_2, ..., a_n$, coefficients of order 2 are $a_{12}, a_{13}, ..., a_{(n-1)n}$, coefficient of order n is $a_{12...n}$. The number of all ANF coefficients equals 2^n.

Let us denote the number of all (zero and non-zero) coefficients of order i of function f as $\sigma_i(f)$. For n-argument function f there are as many coefficients of a given order as there are i-element combinations in n-element set, i.e. $\sigma_i(f) = \binom{n}{i}$.

Hamming Distance

Hamming weight of a binary vector $x \in \Sigma^n$, denoted as $hwt(x)$, is the number of ones in that vector.

Hamming distance between two Boolean functions f, g: $\Sigma^n \rightarrow \Sigma$ is denoted by $d(f, g)$ and is defined as follows:

$$d(f,g) = \sum_{x \in \Sigma^n} f(x) \oplus g(x)$$

The distance of a Boolean function f from a set of n-argument Boolean functions X_n is defined as follows:

$$\delta(f) = \min_{g \in X_n} d(f,g)$$

where $d(f, g)$ is the Hamming distance between functions f and g. The distance of a function f from a set of affine functions A_n is the distance of function f from the nearest function $g \in A_n$.

The distance of function f from a set of all affine functions is called the nonlinearity of function f and is denoted by N_f.

SAC

A Boolean function f satisfies SAC if complementing any single input bit changes the output bit with probability of 0,5.

A Boolean function $f(x_1, ..., x_n)$ satisfies SAC (the strict avalanche criterion) if $f(x) \oplus f(x \oplus \alpha)$ is balanced for any $\alpha \in \Sigma^n$ such that $hwt(\alpha) = 1$.

Exercise:

1. Open the file sbox.sbx in binary editor and read the functions written in it.
2. Check the balance of each function. Is this feature important from a cryptographic point of view? Explain why this is important.
3. Determine the nonlinearity of this functions. To do so, generate the set of all 8-argument affine functions. What is the size of this set?
4. Verify that the strict avalanche criterion (SAC) is satisfied for each function. What value of the probability of change in the output was obtained for the entire block?
5. Write a short report from the class. It should include the obtained results: nonlinearity of the block, yes/no balancing, SAC, description of the method of generating the set of affine functions, summary.