INF 240 - Exercise problems - 12

Nikolay Kaleyski

1 Cryptographic security of vectorial Boolean functions

One of the most important practical applications of (n, m)-functions is in block ciphers in cryptography, where they are employed in the role of so-called "substitution boxes" or "S-boxes". In order to make their implementation and analysis manageable, block ciphers typically incorporate only one non-linear component, which is represented as an (n, m)-function. The security of the entire cipher then directly depends on the properties of the chosen function. Using an S-box with suboptimal cryptographic properties leads to cryptographically weak ciphers that can be broken by various kinds of cryptanalytic attacks. Identifying (n, m)-functions with good cryptographic qualities is thus a crucial part of the design and analysis of block ciphers, and in this assignment we will take a look at how the cryptographic strength of (n, m)-functions can be measured.

In general, cryptanalytic attacks exploit predictable patterns and regularities in the behaviour of functions. Different attacks exploit different weaknesses, and consequently cryptographers have identified different properties and statistics that quantify the resistance of functions to such attacks. We will focus on two of the most powerful known attacks against block ciphers, viz. differential cryptanalysis and linear cryptanalysis.

1.1 Differential uniformity

Differential cryptanalysis exploits dependencies between the difference in the inputs and the outputs to a function. In other words, suppose that $F(x)$ is an (n, m)-function for some n and m. If we have two inputs, x_1 and x_2, let us denote their difference by $d = x_1 - x_2$. The difference between the outputs would be $e = F(x_1) - F(x_2)$. Ideally, if the attacker knows the difference in inputs, this should not give him any information about the outputs; in particular, he should be unable to predict the difference of the outputs any better than just guessing it at random. But if for some input difference d some output difference e occurs frequently (meaning that for a lot of the inputs $(x_1, x_2), (x_3, x_4)$, etc. with input difference $d = x_1 - x_2 = x_3 - x_4$, etc. the output difference is e), the attacker can use this to his advantage.

To express this formally, we define the numbers $\delta_F(a, b)$ which count the number of pairs with a given input difference that map to a given output difference for a given (n, m)-function F. Formally, we define this as

$$\delta_F(a, b) = |\{x \in \mathbb{F}_{2^n} : F(a + x) - F(x) = b\}| \tag{1}$$

where $a \in \mathbb{F}_{2^n}$ and $b \in \mathbb{F}_{2^m}$; equivalently, this is the number of solutions x to the equation $F(a+x) - F(x) = b$. Here a is the input difference and b is the output difference. In order for the function to be secure, there should be no combination (a, b) of input and output difference for which this number of solutions is large. So we define the quantity

$$\delta_F = \max_{0 \neq a \in \mathbb{F}_{2^n}, b \in \mathbb{F}_{2^m}} \delta_F(a, b) \tag{2}$$

which we call the *differential uniformity* of F. The lower the value of δ_F, the more resistant F is to differential cryptanalysis.

Before continuing, we note a few conventions. First, we are in a field of characteristic two, so addition and subtraction are the same; thus, in equation (1), we can replace the minus signs with plus signs. Second, the functions of the form $D_a F(x) = F(a+x) - F(x) = F(a+x) + F(x)$ are used very frequently in the cryptographic analysis of (n, m)-functions, and are called the *derivatives* of F. If F is an (n, m)-function, then so are its derivatives; hence, they can be expressed as truth tables, univariate polynomials, in ANF, etc.

For example, if $F(x) = x^{10}$ is the univariate form of an (n, m)-function, then $D_a F(x) = (x+a)^{10} + x^{10}$ for any $a \in \mathbb{F}_{2^n}$. Using the "freshman's theorem", we can simplify this as

$$(x+a)^8(x+a)^2 + x^{10} = (x^8 + a^8)(x^2 + a^2) + x^{10} = \\ x^{10} + x^8 a^2 + x^2 a^8 + a^{10} + x^{10} = x^8 a^2 + x^2 a^8 + a^{10}.$$

Exercise 1. *Let α be a primitive element of $\mathbb{F}_{2^4}$ and consider the $(4, 4)$-function*

$$F(x) = x^9 + \alpha x.$$

1. *Find the univariate form of the derivative $D_a F(x)$, where a is some element $\mathbb{F}_{2^4}$.*
2. *Find the univariate form of the derivative of $D_a F(x)$ in direction b, i.e. the so-called second derivative $D_b D_a F(x)$, for some element $b \in \mathbb{F}_{2^4}$.*
3. *Find and compare the algebraic degrees of F, $D_a F$ and $D_b D_a F$.*

In general, the only way to compute δ_F for some given (n, m)-function F is to go through all possible values of $a \neq 0$ and b and find the number of solutions to $F(x+a) + F(x) = b$. For small values of n and m, this can be done fairly quickly on a computer. Nonetheless, computing the differential uniformity is easier for *power functions*, i.e. for functions of the form x^d, and for *quadratic functions*, i.e. for functions of algebraic degree 2. In the case of a power function F, the derivatives $D_a F$ "behave" the same with respect to the number of solutions for all non-zero values of a, so it is enough to take any fixed value of a, say $a = 1$, and go through all b. In the case of quadratic functions, it suffices to find the number of solutions x to $D_a F(x) = F(a)$ for all values of a.

Exercise 2. *The following truth table corresponds to a $(3, 3)$-power function. For each three-dimensional binary vector $b \in \mathbb{F}_{2^3}$, find the number of solutions to $F(a+x) + F(x) = b$ for $a = (0, 0, 1)$. What is the differential uniformity of this function?*

x	$F(x)$
000	000
001	101
010	110
011	111
100	100
101	011
110	001
111	010

Table 1: Truth table for a $(3,3)$-power function

It is not difficult to see that the differential uniformity of an (n,m)-function is always even since if x is a solution to $D_aF(x) = F(x+a)+F(x) = b$ for some a and b, then so is $x+a$, since we have

$$D_aF(a+x) = F(a+x+a) + F(a+x) = F(x) + F(a+x) = b.$$

Thus, the lowest possible value of δ_F is 2. The functions which possess this optimal value of δ_F are called *almost perfect nonlinear (APN)*, and as a consequence provide the best possible resistance to differential cryptanalysis.

Exercise 3. *It is known that the so-called Gold functions $G(x) = x^{2^i+1}$ are APN (n,n)-functions for any n with $\gcd(i,n) = 1$. Show that the Gold function for $i = 1$, i.e. $G(x) = x^3$, is APN for any n by the following steps:*

1. *Find the univariate representation of the derivative $D_aF(x)$, where $a \neq 0$.*
2. *Show that $D_aF(x) = b$ cannot have more than 2 solutions for any $b \in \mathbb{F}_{2^n}$.*

1.2 Nonlinearity

Another powerful attack is linear cryptanalysis, which attempts to approximate an (n,m)-function F by an affine (possibly linear) (n,m)-function L. The rationale behind this is that affine (and hence also linear) functions behave in a predictable way and are easy to analyze (in fact, affine functions are by far the worst choice for a cryptographic S-box). Although the function used in a block cipher may not be affine itself, it may be "close" to an affine function (in terms of Hamming distance). The closer F is to an affine function, the more efficient the approximation, and hence the more powerful the attack.

In fact, not only must F itself be far away from all (n,m)-affine functions, but all of its component functions (which are Boolean, i.e. $(n,1)$-functions) must be far away from all $(n,1)$-affine functions. The minimum distance between a Boolean function and all affine functions of the same dimension is called its *nonlinearity*. Similarly, the minimum nonlinearity of any of the component functions of an (n,m)-function F is the *nonlinearity* of F.

A natural question is, how do we find the set of all affine Boolean functions of a given dimension. Fortunately, this is easy to do. We know that an affine function is either a linear function, or a linear function plus a constant. Thus, the problem of finding all affine function is reduced to that of finding all linear functions. Computing the set of all linear functions can be approached in two ways:

- recall that all linear $(n,1)$-functions can be expressed using the trace function as $\mathrm{Tr}_n(ax)$ for $a \in \mathbb{F}_{2^n}$. Thus, all affine functions are of the form $\mathrm{Tr}_n(ax)+b$ for any combination of $a \in \mathbb{F}_{2^n}$ and $b \in \mathbb{F}_2$.
- Recall that by definition a linear function l must satisfy $l(x)+l(y) = l(x+y)$ for any $x, y \in \mathbb{F}_2^n$. Thus, if $B = \{b_1, b_2, \ldots, b_n\}$ is a basis of the vector space $\mathbb{F}_2^n$ and we know the values of l at $b_1, b_2, \ldots, b_n$, we can uniquely reconstruct the truth table of l. For example, if $n = 3$ and we know the values $l(0,0,1)$, $l(0,1,0)$, and $l(1,0,0)$, we can find all remaining values of l.

Exercise 4. *Consider the $(2,1)$-function f given by the truth table*

x	$f(x)$
00	1
01	0
10	1
11	1

Table 2: Truth table of a $(2,1)$-function

Compute its nonlinearity by the following steps:

1. *Find the truth tables of the four linear $(2,1)$-functions by filling in the blanks in Table 3 below. To do this, for example, write all possible combinations of 0's and 1's for the values of the basis $(0,1)$, $(1,0)$, and use them to compute the remaining values of the functions.*
2. *Add the constant 1 to each output value of each linear function to obtain the remaining four affine functions.*
3. *Compute the Hamming distance between each of the eight affine functions and the function $f(x)$ given in Table 2.*

x	$l_1(x)$	$l_2(x)$	$l_3(x)$	$l_4(x)$	$a_1(x)$	$a_2(x)$	$a_3(x)$	$a_4(x)$
00	-	-	-	-	-	-	-	-
01	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-

Table 3: Affine $(2,1)$-functions

Clearly, the higher the non-linearity, the better the resistance against linear cryptanalysis. However, the question off the optimal value of the non-linearity is a bit more difficult than in the case of differential uniformity. We have the so-called *covering radius bound*, which states that the non-linearity N_F of an (n,m)-function satisfies

$$N_F \leq 2^{n-1} - 2^{n/2-1},$$

and the *Sidelnikov-Chabaud-Vaudeney (SCV) bound*, which bounds the nonlinearity of any (n,m)-function with $m \geq n-1$ by

$$N_F \leq 2^{n-1} - \frac{1}{2}\sqrt{3 \cdot 2^n - 2 - 2\frac{(2^n-1)(2^{n-1}-1)}{2^m-1}}.$$

Functions attaining the covering radius bound with equality are called *bent*, but exist only when $m \leq n/2$. Functioning meeting the SCV bound with equality are called *almost bent (AB)* and exist only for $m = n$ with n odd. Note that despite the name, almost bent functions are not inferior to bent functions; the two classes of functions exist under different conditions.

Furthermore, there is a bound on the algebraic degree of AB Functions in terms of the dimension of the finite field, viz. $\deg(F) \leq (n+1)/2$ if F is AB. This is rather unfortunate because the algebraic degree is another useful indicator of cryptographic strength; more precisely, a high algebraic degree indicates a good resistance to higher order differential cryptanalysis.

Exercise 5. *Consider all* $(7,7)$*-functions.*

1. *Compute the value of the covering radius bound for this case.*
2. *Compute the value of the SCV bound for this case.*
3. *Deduce the nonlinearity of a* $(7,7)$*-AB function.*
4. *What is the highest algebraic degree of a* $(7,7)$*-AB function?*

2 Equivalence relations

Since the number of (n,m)-functions grows exponentially with the values of n and m, various notions of equivalence are introduced in order to make their classification manageable. More precisely, equivalence relations between (n,m)-functions are defined, and functions are only classified up to equivalence. Of course, this only makes sense if the equivalence relations in question preserve the cryptographic properties of the functions. All of affine equivalence, extended affine equivalence, EAI-equivalence, cyclotomic equivalence, and CCZ-equivalence preserve both differential uniformity and non-linearity. Other properties of functions may or may not remain invariant under these equivalence relations: for instance, EA-equivalence preserves algebraic degree, whereas CCZ-equivalence does not.

This has a few practical implications for working with functions. First, equivalence relations can be used to derive new functions from known ones. For example, since CCZ-equivalence preserves differential uniformity but not algebraic degree, searching through the equivalence class of an APN function of low algebraic degree is a good way of finding another APN function of high algebraic degree (which, as we mentioned previously, provides good resistance to higher order differential attacks). On the other hand, if a new APN function is discovered, it must be compared for equivalence against all previously known functions.

The most general known equivalence relation which preserves differential uniformity is *CCZ-equivalence*. We say that two (n,m)-functions F and G are CCZ-equivalent if there is an affine permutation which maps the graph $\{(x, F(x)) : x \in \mathbb{F}_{2^n}\}$ of F to the graph of G.

In general, checking whether two given functions are equivalent is quite difficult to do. It is much easier to do for cyclotomic equivalent than, for instance, for CCZ-equivalence. Fortunately, in some particular cases, these two notions of equivalence coincide. This is the case for power functions: two power functions are CCZ-equivalent if and only if they are cyclotomic equivalent.

Recall that two (n, n)-power functions x^d and x^e are *cyclotomic equivalent* if $d = 2^i e \mod (2^n - 1)$ or if $\gcd(e, 2^n - 1) = 1$ and $d = 2^i/e \mod (2^n - 1)$.

Exercise 6. *Consider the following five* $(7, 7)$*-functions:*

$$f_1(x) = x^{10} \quad f_2(x) = x^{23} \quad f_3(x) = x^{25} \quad f_4(x) = x^{40} \quad f_5(x) = x^{101}$$

Partition them into CCZ-equivalence classes, i.e. find all pairs among the given functions that are CCZ-equivalent.