Symmetric key cryptography and its relation to graph theory

E. Pasalic

A modern cryptology relies on many disciplines such as information theory, computer science, probability theory, number theory and abstract algebra. An information theoretical foundation of modern cryptology was established in the late forties. In his celebrated paper [9] from 1948 Claude E. Shannon laid the theoretical foundations of information theory. One of the greatest contribution of his work was a new concept of measuring the information. In his second work [10], among other important notion, Shannon introduced the concept of *unconditional security* of symmetric ciphers. Unconditional security means that even if an adversary is assumed to have unlimited computational resources he still cannot defeat the cryptosystem. A necessary condition for a symmetric-key encryption scheme to be unconditionally secure is that the encryption key is at least as long as the message, which obviously restricts the practical use of such a system. Also, Shannon introduced two extremely important concepts which have been extensively used in design of modern ciphers, namely *confusion* and *diffusion.*

A standard cryptosystem model used for achieving confidentiality (secrecy), also called symmetric-key cryptosystem transforms the plaintext message m into the ciphertext message c so that $c = E_K(m)$, where E_K denotes the encryption function, see Figure 1.

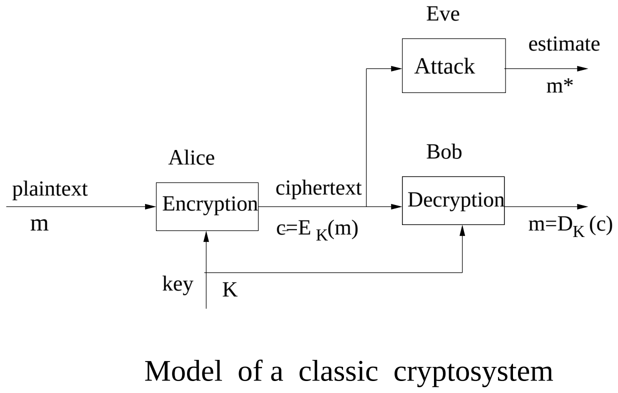

Figure 1: Symmetric-key cryptosystem

The ciphertext message received by Bob is now supposed to be decrypted before reading. Equipped with the same key as Alice, Bob performs the following. He applies the decryption algorithm D_K on the encrypted message, i.e., $m = D_K(E_K(m))$ and retrieves the original message. The cryptanalyst Eve, not knowing the actual key K, may perform various attacks on the cryptosystem. The most trivial one, is called *exhaustive search* which checks for all possible keys in the key space to decrypt the message.

As an example of an insecure symmetric-key cryptosystem we consider the Vigenère cipher. It is assumed that both the message and key symbols are letters from the English alphabet, i.e., $\mathcal{M}, \mathcal{K} \in \{A, B, \ldots, Z\}$. A sequence of message symbols $\mathbf{m} = m_0, m_1, \ldots$ is encrypted by this scheme into an encrypted sequence $\mathbf{c} = c_0, c_1, \ldots$ as follows. In order to express the encryption mathematically a simple transformation is performed, namely the letters are replaced by integers such that, $A \leftrightarrow$

$0, B \leftrightarrow 1, \ldots, Z \leftrightarrow 25$. The same transformation is applied to the key $\mathbf{K} = K_0, K_1, \ldots, K_{l-1}$ and the corresponding message and key sequence are denoted $\mathbf{m}'$ and $\mathbf{K}'$, respectively. Then, the encrypted integer sequence $\mathbf{c}' = c'_0, c'_1, \ldots$ is obtained using,

$$c'_i = m'_i + K'_{i \bmod l} \bmod 26, \quad i = 0, 1, 2, \ldots. \tag{1}$$

Now the ciphertext $\mathbf{c}$ is derived from $\mathbf{c}'$ using the reverse transformation, $0 \leftrightarrow A, 1 \leftrightarrow B, \ldots, 25 \leftrightarrow Z$. To recover the sequence of the original message, a similar transformation is applied to the encrypted sequence by the recipient,

$$m'_i = c'_i + (26 - K'_{i \bmod l}) \bmod 26, \quad i = 0, 1, 2, \ldots.$$

Then the same transformation as above is applied to $\mathbf{m}'$ to retrieve the sequence of alphabetic letters $\mathbf{m}$.

Nevertheless, practical encryption schemes use more sophisticated approaches of implementing Shannon's concepts of confusion and diffusion. The encryption is rather performed on a bit level (or on a block of bits) by either ”expanding” the secret key of finite length into a pseudo random sequence (running key sequence) z_i using keystream generator (*stream ciphers*), see Figure 2.

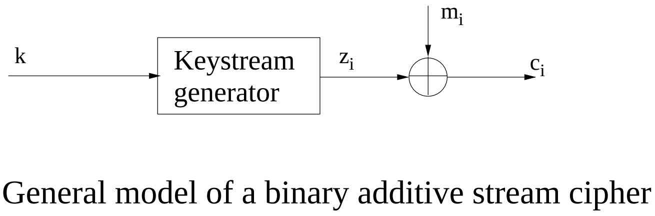

Figure 2: Additive (binary) stream cipher

Alternatively, an encryption scheme can be designed by implementing a pseudo random permutations that substitutes a block of data (typically 128 bits) by a block of ciphertext bits of the same length (*block ciphers*) by repeating substitution (S) and permutation (P) through sevral rounds, see Figure 3.

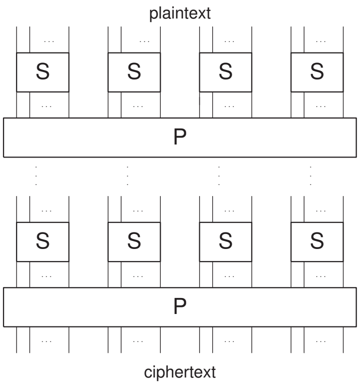

Figure 3: Substitution permutation network using S-boxes - a block cipher

In both cases an essential cryptographic primitive for embedding the concept of confusion is so-called Boolean function. Denoting by $\mathbb{F}_2$ the binary Galois field (thus $\mathbb{F}_2 = \{0, 1\}$) and the n-dimensional vector space over $\mathbb{F}_2$ by $\mathbb{F}_2^n$, a Boolean function is defined as $f : \mathbb{F}_2^n \to \mathbb{F}_2$. A vectorial Boolean function $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$, also known as substitution box (S-box), is widely used primitive in the design of block ciphers. For instance, the S-boxes of DES (Data Encryption Standard) use $F : \mathbb{F}_2^6 \mapsto \mathbb{F}_2^4$, whereas the new standard AES (Advanced Encryption Standard) use $F : \mathbb{F}_2^8 \mapsto \mathbb{F}_2^8$. Since S-boxes are commonly the only nonlinear components of the block cipher, their design is crucial from the security point of view.

LFSR based stream ciphers and basic definitions

Stream ciphers which make use of a Boolean function are classically divided into two major groups: *nonlinear combination generator* and *nonlinear filter generators*, see Figure 4.

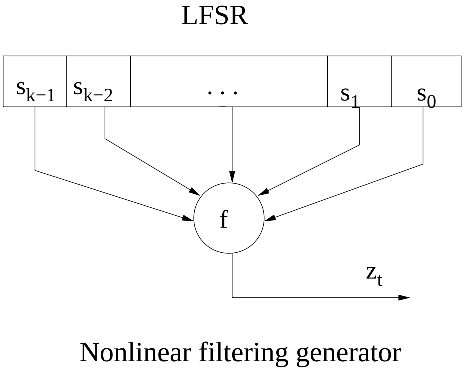

Figure 4: Nonlinear filtering generator

Both schemes have in common the use of a *linear feedback shift register* (LFSR) as a main constituent block for producing sequences of large period. LFSRs are very well suited for hardware implementation and they can produce sequences with very good statistical properties. In relation to Figure 5, the update procedure performed in any LFSR (at the time instance controlled by the system clock) may be summarized as follows :

1. The content of stage 0 is output and forms a part of the output sequence s_i, and at the same time the new content of stage $k-1$ is computed using a linear recursion $s_k = \sum_{i=0}^{k-1} s_i c^{k-i}$.

2. The content of stage i is moved to stage $i-1$, for each $1 \leqslant i \leqslant k-1$. The next state of the LFSR is therefore $S = (s_k, \ldots, s_1)$ seen from left to right in Figure 5.

For a given length of the LFSR, the period and statistical properties of the sequence depend entirely on the *connection polynomial* used. The use of a primitive connection polynomial $c(x) \in \mathbb{F}_2[x]$ results in the *sequence of maximum length* (the length is $2^L - 1$ for an LFSR of length L) with good statistical properties. Informally, a primitive polynomial $p(x) = a_0 + a_1 x + \ldots + a_k x^k$ of degree k can be defined as an irreducible polynomial over $\mathbb{F}_2$ with the property that $\{x^i \pmod{p(x)} : i = 0, \ldots, 2^k - 2\} = \mathbb{F}_2^k \setminus \{0\}$, using the representation $x^i \pmod{p(x)} = r(x) = r_0 + r_1 x + \ldots + r_{k-1} x^{k-1}$ and identifying $(r_0, \ldots, r_{k-1})$ with the elements of $\mathbb{F}_2^k$.

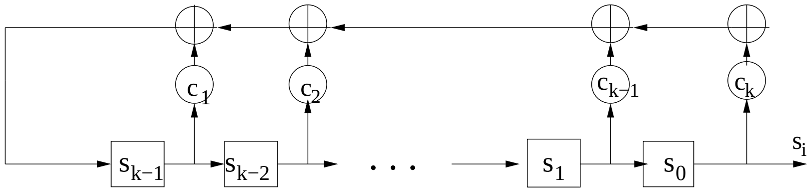

Figure 5: LFSR of length k with connection polynomial

Let s denote an infinite binary sequence whose terms are $s_0, s_1, \ldots$, whereas its truncated version of finite length n is denoted by s^n, that is, $s^n = s_0, s_1, \ldots, s_{n-1}$. The following definitions, taken from [6], will be useful in the sequel.

Definition 1 *An LFSR is said to* generate *a sequence s if there is some initial state of LFSR for which the output sequence of the LFSR is s. Similarly, an LFSR generates s^n if for some initial state the first n terms of the output sequence of the LFSR coincide with s^n.*

Definition 2 *The* linear complexity *of an infinite binary sequence s, denoted $L(s)$, is the length of the shortest LFSR that generates s.*

Example 1 *For $k = 4$ (or L $= 4$) and the primitive connection polynomial $C(x) = x^4 + x + 1$ if we start the LFSR with $S = (s_0, s_1, s_2, s_3) = (1, 1, 1, 0)$ it produces the sequence*

$$1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1|1, 1, 1, 0 \ldots$$

The sequence is of maximum length $15 = 2^4 - 1$ and contains exactly $2^{k-1} = 8$ ones and $2^{k-1} - 1$ zeros, why ? Check what happens if we use irreducible polynomial $C(x) = x^4 + x^3 + x^2 + x + 1$!

However, any sequence generated by a finite-state machine has a finite linear complexity. Moreover, due to Elwyn R. Berlekamp and James L. Massey [5], there exists an efficient polynomial-time synthesis algorithm, which computes the linear complexity of a given binary sequence. When the length L of LFSR is known then a sequence of length $2L$ is required to compute the connection polynomial, either using the Berlekamp-Massey algorithm or a direct matrix equation. If L is not known, then the Berlekamp-Massey algorithm can be used to determine L and the connection polynomial. In either case the adversary must obtain a subsequence of length $2L$.

In reference to Figure 2, we assume that an adversary mounts a known or chosen-plaintext attack on additive binary stream cipher where the running-key generator is implemented by using an LFSR. Then the adversary can obtain the subsequence of $\mathbf{z}$ of length L, by computing $z_i = m_i \oplus c_i$, $i = 0, \ldots, L - 1$ (since m_i are known). Then, an LFSR of length L, with the connection polynomial computed with the Berlekamp-Massey algorithm, can be initialized with this subsequence to generate the remainder of the sequence $\mathbf{z}$.

Thus, a necessary but not sufficient condition for any keystream generator is the requirement for a large linear complexity. This cannot be achieved using a single LFSR, and general methods for destroying the linear properties of LFSRs are:

- using a *nonlinear combining function* at the outputs of several LFSRs;
- using a *nonlinear filtering function* on the contents of a single LFSR;and
- using the output of one/several LFSRs to control clocking of one/several LFSRs.

As mentioned earlier the first two methods take advantage of a Boolean function to introduce the nonlinearity to the keystream. A general construction of a nonlinear combination generator is illustrated in Figure 6, where for the sake of generality we consider $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$, for $m \geqslant 1$.

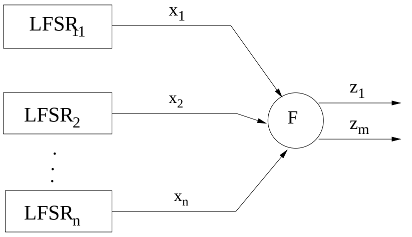

Figure 6: Nonlinear combination generator

In this set up the outputs of n LFSRs, $x^{(1)}, \ldots, x^{(n)}$ are used as the inputs to a nonlinear vectorial Boolean function, denoted F, and the keystream sequence is then generated by this function. More formally, $z_i \overset{\triangle}{=} f_i(x_i^{(1)}, \ldots, x_i^{(n)})$, and the function $F : \mathbb{F}_2^n \mapsto \mathbb{F}_2^m$ (actually an S-box) is a collection of m Boolean functions $F = (f_1, \ldots, f_m)$. A Boolean function $f(x_1, \ldots, x_n)$ can be represented as the output column of its *truth table* f, i.e., a binary string of length 2^n, $f = [f(0, 0, \cdots, 0), f(1, 0, \cdots, 0), f(0, 1, \cdots, 0), \ldots, f(1, 1, \cdots, 1)]$.

The truth table representation may be suitable for Boolean function in small number of variables. Thus, for moderate to large values of n, $f \in \mathcal{B}_n$ is usually represented by its *algebraic normal form* (ANF):[1]

$$f(x_1, \ldots, x_n) = \sum_{u \in \mathbb{F}_2^n} \lambda_u \left(\prod_{i=1}^{n} x_i^{u_i} \right) \ , \quad \lambda_u \in \mathbb{F}_2 \ , u = (u_1, \ldots, u_n). \qquad (2)$$

There are 2^n different terms $x_1^{u_1} x_2^{u_2} \cdots x_n^{u_n}$ for different u's. As λ_u is binary it gives $\#\mathcal{B}_n = 2^{2^n}$ different functions in n variables $x_1, \ldots, x_n$ (denoting by $\mathcal{B}_n$ the set of all Boolean functions in n variables), implying that a search for ”good” functions becomes infeasible already for $n = 6$!

Example 2 *For $n = 3$ there are $2^8 = 256$ distinct functions specified by λ_u,*

$$\mathcal{B}_3 = \{\lambda_0 1 \oplus \lambda_1 x_1 \oplus \lambda_2 x_2 \oplus \lambda_3 x_3 \oplus \lambda_4 x_1 x_2 \oplus \lambda_5 x_1 x_3 \oplus \lambda_6 x_2 x_3 \oplus \lambda_7 x_1 x_2 x_3\}.$$

The *algebraic degree* of f, denoted by $deg(f)$ or sometimes simply d, is the maximal value of the Hamming weight of u such that $\lambda_u \neq 0$. There is a one-to-one correspondence between the truth table and the ANF via so called inversion formulae.

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

The truth table of the Boolean function $f(x_1, x_2, x_3) = x_1 x_2 + x_2 x_3 + x_3$.

The easiest way to obtain the ANF from the truth table (without involving Möbius transform) is to expand the ANF of f when $f(x) = 1$ and add these together. For the above example we have:

$$f(x) = x_1 x_2 (1 + x_3) + (1 + x_1)(1 + x_2) x_3 + x_1 (1 + x_2) x_3 + x_1 x_2 x_3 = x_1 x_2 + x_2 x_3 + x_3,$$

after cancelling identical terms. A *balanced* Boolean function has equally many zeros and ones in its truth table, i.e., $\{f(x) = 0 : x \in \mathbb{F}_2^n\} = \{f(x) = 1 : x \in \mathbb{F}_2^n\} = 2^{n-1}$. What can be said about the upper bound on degree of balanced Boolean functions in $\mathcal{B}_n$ then ?

The reason why we require a high algebraic degree is related to the following attack scenario. Recall that the basic goal of the attacker is to recover the secret state bits located in LFSR. Since both LFSR, its connection polynomial $c(x)$, the filtering function $f(x)$ and a portion of the output keystream sequence (known-plaintext attack) are known we have the following. At each time

[1] Addition operator over $\mathbb{F}_2$ denoted by “$\oplus$” is often replaced with usual addition operator “$+$”.

instance the known keystream bit $z_i^t = f(x_1^t, \ldots, x_n^t)$, where the time dependency of the inputs to f is due to the structure of LFSR. Anyway, any x_i^t is a linear function of the initial secret state bits $s_0, \ldots, s_{L-1}$, say $x_i^t = \sum_{j=0}^{L-1} a_j^{(i,t)} s_j$, due to the linear update function of LFSR. Thus given f of degree d, whose ANF contains at most $T = \binom{n}{0} + \binom{n}{1} + \ldots + \binom{n}{d}$ terms, we get one equation of degree d is secret state bits. Using so-called linearization we can introduce (at most) T new variables in $s_0, \ldots, s_{L-1}$ and solve a linear system with respect to unknown and secret s_i. Since there are L secret state variables after the above substitution our linear system has at most $T' = \binom{L}{0} + \binom{L}{1} + \ldots + \binom{L}{d}$ terms. The complexity of solving a linear system of size $\approx \binom{L}{d}$ is of order $(\binom{L}{d})^3$ using Gauss elimination. Therefore, a large d is desirable but the implementation cost increases !

Assume now, that n maximum-length LFSRs as in Figure 6, whose lengths $L_1, L_2, \ldots, L_n$ are relatively prime, are combined by a nonlinear Boolean function $f(x_1, \ldots, x_n)$. Then the linear complexity of the keystream sequence z is $f(L_1, \ldots, L_n)$, where the expression is computed over the integers [6, 12]. Since this expression is directly dependent on the degree of f, then obviously a large linear complexity of the keystream is obtained by functions of high degree.

Example 3 *(*Geffe generator*) Assume that the lengths of LFSRs are relatively prime for the scheme in Figure 6, with $n = 3$. Let the nonlinear combining function be $f(x_1, x_2, x_3) = x_1x_2 \oplus x_2x_3 \oplus x_3$. The function f is obviously of degree 2. The Geffe generator is cryptographically weak because the information about the states of LFSR$_1$ and LFSR$_3$ leaks to the output. For fixed $x_3 = 0$ the output is x_1x_2 and therefore 75% zeros and 25% of ones are outputted in this case.*

The observation in the above example leads to another important criteria for Boolean functions used as a nonlinear combining function, which is the concept of *correlation immunity.*

Definition 3 *[11] Let $x_1, x_2, \ldots, x_n$ be a set of independent uniformly distributed binary random variables. A Boolean function $f(x_1, x_2, \ldots, x_n)$ is called mth order* correlation immune *if for each subset of at most m input variables $x_{i_1}, \ldots, x_{i_k}$, $1 \leqslant i_1 \cdots \leqslant i_k \leqslant n$, $k \leqslant m$, the mutual information between the keystream $z = f(x_1, \ldots, x_n)$ and the subset $x_{i_1}, \ldots, x_{i_k}$ is equal to zero, i.e. $I(z; x_{i_1}, \ldots, x_{i_k}) = 0$. Expressed in terms of probability we have that*

$$Prob(x_{i_1} \oplus x_{i_2} \cdots \oplus x_{i_k} = z) = \frac{1}{2}, \quad z \in \mathbb{F}_2, \quad for\ any\ \ k = 1, \ldots, m.$$

Another important measure of cryptographical strength of Boolean functions is *nonlinearity.* The nonlinearity of f, denoted by $\mathcal{N}_f$, is defined to be the minimum Hamming distance [2] to the set of affine functions. For an n-input variable function the set of affine functions is given as $\mathcal{A}_n = \{a_1x_1 \oplus \cdots \oplus a_nx_n \oplus b,\ a \in \mathbb{F}_2^n;\ b \in \mathbb{F}_2\}$. The set of all n variable linear functions, when $b = 0$, is denoted by $\mathcal{L}_n$. Thus, the nonlinearity of f is given by,

$$\mathcal{N}_f = \min_{g \in \mathcal{A}_n} d_H(f, g). \tag{3}$$

Prof. James Massey formulated it nicely once upon a time " The linearity is the curse of the cryptographer". Any cryptographic primitive somehow implements Shannon's concept of confusion which for our scheme (almost) directly corresponds to nonlinearity.

The linear functions will be represented by means of the scalar (inner) product, $\varphi_\alpha : x \in \mathbb{F}_2^n \longmapsto \alpha \cdot x = \sum_{i=1}^n \alpha_i x_i$.

Definition 4 *A t-th order correlation immune function Boolean function f which is balanced is called a t-*resilient *function.*

[2] The Hamming distance between two binary strings of the same length, say f and g, is the number of positions where these strings differ, i.e., $d_H(f, g) = \#\{x | f(x) \neq g(x)\}$.

The properties of Boolean functions are most comprehensibly viewed through the *Walsh transform*.

Definition 5 *The* Walsh transform *of $f \in \mathcal{B}_n$ in point $\alpha \in \mathbb{F}_2^n$ is denoted by $\mathcal{F}(f + \varphi_\alpha)$ and calculated as,*

$$\alpha \in \mathbb{F}_2^n \longmapsto \mathcal{F}(f + \varphi_\alpha) = W_f(\alpha) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) + \varphi_\alpha(x)} . \tag{4}$$

The values of these coefficients form the *Walsh-spectrum* of f, and clearly f is balanced if and only if $W_f(0) = 0$. Notice that $\varphi_\alpha(x) = \alpha \cdot x$ uniquely identifies one linear function, see also relation (5).

Exercise 1 *Show that the Hamming distance between a Boolean function $f(x)$ and an affine function $g(x) = \alpha \cdot x + b$ ($\alpha \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2$), can be calculated via the Walsh transform as $d_H(f, g) = 2^{n-1} - \frac{(-1)^b \mathcal{F}(f+\varphi_\alpha)}{2}$.*

A closely related concept, known as the Hadamard transform and denoted by W_f^H, simply uses the values $f(x)$ instead of $(-1)^{f(x)}$, that is $W_f^H(\alpha) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{\varphi_\alpha(x)}$. A simple relationship between the two transforms is given as an exercise.

Exercise 2 *Show that $W_f(\alpha) = -2W_f^H(\alpha) + 2^n \Delta(\alpha)$ for any $\alpha \in \mathbb{F}_2^n$, where $\Delta(\alpha) = 1$ if $\alpha = 0$, and zero otherwise.*

The values of Walsh and Hadamard spectra of $f \in \mathcal{B}_n$ are easily obtained through $W_f = H_n f^T$, respectively, $W_f^H = H_n(-1^f)^T$, where f^T denotes the transpose of the truth table of f and H_n is the Hadamard matrix of size $2^n \times 2^n$ defined recursively,

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad H_n = \begin{pmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{pmatrix}.$$

It is easy to show that $HH^T = 2^n I$ and also $H^T H = 2^n I$, where I is the identity matrix whose diagonal elements are ones.

The nonlinearity of $f(x)$ can be obtained via the Walsh transform as,

$$\mathcal{N}_f = 2^{n-1} - \frac{1}{2} \max_{\alpha \in \mathbb{F}_2^n} |\mathcal{F}(f + \varphi_\alpha)|. \tag{5}$$

Lemma 1 *[13] Let $f \in \mathcal{B}_n$ and let t be some positive integer. The function f is said to be* correlation immune *(CI) of order t if and only if $\mathcal{F}(f + \varphi_\alpha) = 0$ for any $a \in \mathbb{F}_2^n$ such that $1 \leqslant wt(\alpha) \leqslant t$.*

An important property of the Walsh spectra, referred to as Parseval's equality [4], states that for any Boolean function $f \in \mathcal{B}_n$, $\sum_{\alpha \in \mathbb{F}_2^n} \mathcal{F}^2(f + \varphi_\alpha) = 2^{2n}$.

Exercise 3 *Use a similar technique as in the proof of Proposition 1 to show Parseval's equality. Consider the sum $\sum_{u \in \mathbb{F}_2^n} W_f(u) W_f(u \oplus v)$ and show it is 2^{2n} if $v = 0$ and zero otherwise.*

We illustrate the cryptographic criteria discussed above with a detailed examination of the nonlinear combining function used in the Geffe generator, see also Example 3.

Example 4 *Consider the function $f(x_1, x_2, x_3) = x_1 x_2 + x_2 x_3 + x_3$ used in the Geffe generator. The truth table and the Walsh spectra are given in Table 1. Note that the linear functions φ_α are determined by x values. For instance the entry $(x_1, x_2, x_3) = (1, 0, 0)$ will yield $\varphi_\alpha = (x_1, x_2, x_3) \cdot (1, 0, 0) = x_1$. Then, the nonlinearity $\mathcal{N}_f = 2^{n-1} - \frac{1}{2} \max_{\alpha \in \mathbb{F}_2^n} |\mathcal{F}(f + \varphi_\alpha)| = 2$. The function is balanced but not correlation immune since $\mathcal{F}(f + x_1) = \mathcal{F}(f + x_3) \neq 0$.*

x_1	x_2	x_3	$f(x)$	$\mathcal{F}(f+\varphi_\alpha)$
0	0	0	0	0
0	0	1	0	-4
0	1	0	0	0
0	1	1	1	-4
1	0	0	1	-4
1	0	1	1	0
1	1	0	0	4
1	1	1	1	0

Table 1: The truth table and the Walsh spectra of the Boolean function $f(x_1, x_2, x_3) = x_1x_2 + x_2x_3 + x_3$.

Notice that the Walsh spectra, constrained by Parseval's equality, is integer valued and obviously we cannot design cryptographically strong Boolean functions by specifying the values (placing zeros and controlling maximum values) in the Walsh spectra (even though Parseval's equality is satisfied). This means that the Boolean space is only a small subspace of a more general mapping from $\mathbb{Z}^n$ to $\mathbb{Z}$.

Proposition 1 *Given the Walsh spectra $\{W_f(\alpha)\}$ of $f \in \mathcal{B}_n$ the inverse Walsh transformation can be computed as,*

$$(-1)^{f(x)} = 2^{-n} \sum_{\alpha \in \mathbb{F}_2^n} W_f(\alpha)(-1)^{\alpha \cdot x} \qquad \text{for all } x \in \mathbb{F}_2^n. \tag{6}$$

Proof. Let us substitute $W_f(\alpha) = \sum_{y \in \mathbb{F}_2^n} (-1)^{f(y)+\alpha \cdot y}$ in $f(x)$ so that,

$$\begin{aligned} \sum_{\alpha \in \mathbb{F}_2^n} W_f(\alpha)(-1)^{\alpha \cdot x} &= \sum_{\alpha \in \mathbb{F}_2^n} \sum_{y \in \mathbb{F}_2^n} (-1)^{f(y)+\alpha \cdot y}(-1)^{\alpha \cdot x} \\ &= \sum_{y \in \mathbb{F}_2^n} (-1)^{f(y)} \sum_{\alpha \in \mathbb{F}_2^n} (-1)^{\alpha \cdot (x+y)} \\ &= 2^n(-1)^{f(x)}, \end{aligned}$$

since since the sum $\sum_{\alpha \in \mathbb{F}_2^n} (-1)^{\alpha \cdot (x+y)}$ is equal to zero unless $x = y$ in which case it is equal to 2^n. The statement follows. □

A special class of functions achieving the upper bound on nonlinearity is known as *bent functions*. They exist only for even n and have a uniform spectra, that is, f is bent if and only if $W_f(\alpha) = \pm 2^{n/2}$, for all $\alpha \in \mathbb{F}_2^n$. It is easily understood that since $\sum_{\alpha \in \mathbb{F}_2^n} W_f(\alpha)^2 = 2^{2n}$, then $\{W_f(\alpha)$ is minimized with respect to its maximum absolute value if the spectra is flat. These functions are not balanced however, since $W_f(0) = \pm 2^{n/2}$, but they posses many other desirable properties and have several connections to difference sets, Kerdock codes, symmetric design etc. (their modified balanced versions are also used in symmetric key primitives). Bent functions correspond to strongly distance regular Cayley graphs, this connection is discussed later.

For any bent function f one may define its dual $\tilde{f}$ as $(-1)^{\tilde{f}(x)} = 2^{-n/2} W_f(x)$ for all $x \in \mathbb{F}_2^n$.

Proposition 2 *The dual bent function $\tilde{f}$ of a bent function f is again bent.*

Proof. If f is bent the inverse Walsh transform gives, $(-1)^{f(x)} = 2^{-n}\sum_{\alpha\in\mathbb{F}_2^n} W_f(\alpha)(-1)^{\alpha\cdot x}$, for all $x \in \mathbb{F}_2^n$. Replacing $W_f(\alpha) = 2^{n/2}(-1)^{\tilde{f}(\alpha)}$ from the definition of $\tilde{f}$, we get

$$2^{n/2}(-1)^{f(x)} = \sum_{\alpha\in\mathbb{F}_2^n}(-1)^{\tilde{f}(\alpha)}(-1)^{\alpha\cdot x} = \sum_{\alpha\in\mathbb{F}_2^n}(-1)^{\tilde{f}+\alpha\cdot x} = W_{\tilde{f}}(\alpha),$$

thus $W_{\tilde{f}}(\alpha) \in \{-2^{n/2}, 2^{n/2}\}$ and $\tilde{f}$ is bent. One class of bent functions of particular importance, known as the Maiorana-McFarland class, is specified as follows. Let us, for $n = 2k$, identify $\mathbb{F}_2^n$ with $\mathbb{F}_2^k \times \mathbb{F}_2^k$. Suppose $\pi : \mathbb{F}_2^k \to \mathbb{F}_2^k$ is a permutation and $g \in \mathcal{B}_k$. A function $f : \mathbb{F}_2^k \times \mathbb{F}_2^k \to \mathbb{F}_2$ defined by

$$f(x,y) = x\cdot\pi(y) + g(y), \text{ for all } x, y \in \mathbb{F}_2^k, \tag{7}$$

is a bent function and this class is denoted as $\mathcal{M}$.

Proposition 3 *The function f defined by (7) is a bent function.*

Proof. The Walsh transform at $(a,b) \in \mathbb{F}_2^k \times \mathbb{F}_2^k$ equals to:

$$W_f(a,b) \quad = \quad \sum_{x\in\mathbb{F}_2^k}\sum_{y\in\mathbb{F}_2^k}(-1)^{f(x,y)+(a,b)\cdot(x,y)} = \sum_{y\in\mathbb{F}_2^k}(-1)^{g(y)+b\cdot y}\sum_{x\in\mathbb{F}_2^k}(-1)^{x\cdot\pi(y)+a\cdot x}.$$

For any fixed y the sum $\sum_{x\in\mathbb{F}_2^k}(-1)^{x\cdot\pi(y)+a\cdot x} = \sum_{x\in\mathbb{F}_2^k}(-1)^{x\cdot(\pi(y)+a)} = 0$, unless $\pi(y) = a$ which happens exactly for one $y = \pi^{-1}(a)$. In the case $\pi(y) = a$ the sum $\sum_{x\in\mathbb{F}_2^k}(-1)^{x\cdot(\pi(y)+a)} = 2^k$, and therefore $W_f(a,b) = 2^k(-1)^{g(\pi^{-1}(a))+b\cdot\pi^{-1}(a)}$, thus f is bent. □

Notice that the class $\mathcal{M}$ contains as a subclass a class of bent functions, but it can also generate resilient functions with high nonlinearity. To see this we modify the above definition as follows,

Definition 6 *For any positive integers p, q such that $n = p+q$, a function $f \in \mathcal{B}_n$ in the Maiorana McFarland class is defined by*

$$f(x,y) = \phi(y)\cdot x \oplus g(y), \quad x \in \mathbb{F}^p,\ y \in \mathbb{F}^q, \tag{8}$$

where ϕ is any mapping from $\mathbb{F}^q$ to $\mathbb{F}^p$, $g \in \mathcal{B}_q$ is arbitrary.

Proposition 4 *Let f be defined as above and for $p > q$ assume that π is injective. Then, $N_f = 2^{n-1} - 2^{p-1}$. In addition, if $wt(\phi(y)) \geqslant t+1$ for all $y \in \mathbb{F}_2^q$ then f is t-resilient.*

Proof. Let $\mathbb{F}_2^n = \mathbb{F}_2^p \times \mathbb{F}_2^q$. All we have to do is to show that $\max_{(a,b)\in\mathbb{F}_2^p\times\mathbb{F}_2^q} \mid W_f(a,b) \mid= 2^p$. We have,

$$W_f(a,b) \quad = \quad \sum_{y\in\mathbb{F}_2^q}\sum_{x\in\mathbb{F}_2^p}(-1)f(x,y) + (a,b)\cdot(x,y) = \sum_{y\in\mathbb{F}_2^q}(-1)^{g(y)+b\cdot y}\sum_{x\in\mathbb{F}_2^p}(-1)\phi(y)\cdot x + a\cdot x.$$

Then again, for any fixed $y \in \mathbb{F}_2^q$ the sum $\sum_{x\in\mathbb{F}_2^p}(-1)\phi(y)\cdot x + a\cdot x = 0$, unless $\pi(y) = a$. Since π is injective then $\#\{y \in \mathbb{F}_2^q : \pi(y) = a\}$ is either 0 or 1. In the case $\pi(y) = a$ we have $\sum_{x\in\mathbb{F}_2^p}(-1)\phi(y)\cdot x + a\cdot x = 2^p$, and the first part follows. The second part is left as an exercise. □

Example 5 *Let $n = 6$, $p = 4$, $q = 2$ and $(x,y) \in \mathbb{F}_2^4 \times \mathbb{F}_2^2$. Define injective $\pi : \mathbb{F}_2^2 \to \mathbb{F}_2^4$ as $\pi(00) =$ (1100), $\pi(10) =$ (0110), $\pi(01) =$ (1010), $\pi(1) =$ (10011). Then, for any fixed y the function $f(x,y)$ is a linear function in $x_1, \ldots, x_4$. More precisely, $f(x,00) = x_1 + x_2$, $f(x,10) = x_2 + x_3$, $f(x,01) = x_1 + x_3$, $f(x,11) = x_1 + x_3 + x_4$. Then f is 1-resilient, deg(f) = 3 (check this !), and $\mathcal{N}_f = 24$.*

More advanced construction methods are not treated here due to their tedious representation. The currently best known methods are given recently by Pasalic and Zhang based on the use of disjoint linear codes (resilient S-boxes) and a subtle modification of the Maiorana-McFarland construction for resilient Boolean functions.

Equivalence classes of Boolean functions

The group of all invertible $\mathbb{F}_2$-linear transformations on $\mathbb{V}_n$ is denoted by $GL(\mathbb{V}_n)$.

Definition 7 *Two Boolean functions $f, g \in \mathcal{B}_n$ are said to be* affine equivalent *if and only if there exist $A \in GL(\mathbb{V}_n)$ and $b \in \mathbb{V}_n$ such that*

$$g(x) = f(Ax + b) \textit{ for all } x \in \mathbb{V}_n. \tag{9}$$

The affine general linear group $AGL(\mathbb{V}_n)$ consists of all the element of the form (A, b). It can be verified that the transformation $f(x) \mapsto f(Ax + b)$ is a group action of $AGL(\mathbb{V}_n)$ on $\mathcal{B}_n$.

Definition 8 *Two Boolean functions $f, g \in \mathcal{B}_n$ are said to be* extended affine equivalent (EA-equivalent, or, equivalent) *if and only if apart from A and b as above there exist $\mu \in \mathbb{V}_n$ and $\epsilon \in \mathbb{F}_2$ such that*

$$g(x) = f(Ax + b) + \mu \cdot x + \epsilon \textit{ for all } x \in \mathbb{F}_2^n. \tag{10}$$

Given any two Boolean functions $f, g \in \mathcal{B}_n$ deciding whether they are EA-equivalent or not is an important open problem. A direct verification requires a search over all the elements of $AGL(\mathbb{V}_n)$ and therefore its computational complexity is $O(2^{n^2})$. Since an exhaustive search over all the elements of $AGL(\mathbb{V}_n)$ is not feasible for $n \geq 7$, the decision problem involving equivalence of Boolean functions is attempted by using carefully chosen invariants. Algebraic degree of a non-affine Boolean function is an invariant with respect to affine transformations and addition of affine functions. Therefore, two Boolean functions with algebraic degree greater than or equal to 2 are EA-inequivalent if their algebraic degrees are different. It is well known [3] that the absolute Walsh spectra of any Boolean function f are invariants with respect to the action of $AGL(\mathbb{V}_n)$ and the addition by an affine function. Unfortunately these invariants are not useful to determine affine inequivalence of Boolean functions having the same algebraic degree and absolute Walsh spectra. The problem of classifying Boolean functions and bent functions in particular seems to be elusive.

Open Problem 1 *Find new classes of bent functions by proving their affine non-equivalence to already known classes. The problem may also be viewed in terms of suitable subgroups of permutations of the Walsh spectra. Indeed, since the dual bent function is also bent it implies that either $\{\alpha : W_f(\alpha) = 2^{n/2}\} = 2^{n-1} - 2^{n/2-1}$ and $\{\alpha : W_f(\alpha) = -2^{n/2}\} = 2^{n-1} + 2^{n/2-1}$ or vice versa. This is also related to a group action on the (multi)set of the Walsh spectra.*

Open Problem 2 *A related concept to the above is so-called* algebraic thickness *which refers to the most compact representation of a function by its ANF. For instance, the function $f(x_1, \ldots, x_n) = x_1 x_2 \cdots x_n$ (which is cryptographically disastrous, why ?) is obviously affine equivalent to the function $f(x_1, \ldots, x_n) = (x_1 + 1)(x_2 + 2) \cdots (x_n + 1)$. While the former contains a single term in its ANF, the latter contains all possible 2^n terms in its ANF. Of course, if we would implement such a function we would prefer the former one. Given any function $f \in \mathcal{B}_n$ find efficiently its affine equivalent containing the least number of ANF terms !*

Vectorial Boolean functions - substitution boxes

The *nonlinearity* of $F = (f_1, f_2, \ldots, f_m)$, denoted by N_F, is defined as the minimum among the nonlinearities of all nonzero linear combinations of the component functions of F, i.e.,

$$nl(F) = \min_{\tau \in \mathbb{F}_2^{m*}} nl(\sum_{j=1}^{m} \tau_j f_j(x)), \text{ where } \tau = (\tau_1, \ldots, \tau_m) \in \mathbb{F}_2^{m*}. \tag{11}$$

The *algebraic degree* of F is defined as the minimum of degrees of all nonzero linear combinations of the component functions of F, namely,

$$deg(F) = \min_{\tau \in \mathbb{F}_2^{m*}} deg(\sum_{j=1}^{m} \tau_j f_j(x)). \tag{12}$$

The two measures defined above in terms of linear combinations of the component functions obviously make the design of cryptographically strong vectorial Boolean functions much harder than in the Boolean case. In certain situations one may use additional algebraic structures in those cases such structures are available, but usually one prefer to involve the structure of finite fields and to consider mappings F over $\mathbb{F}_{2^n}$ so that isomorphically $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is equivalent to $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ (once the basis of the finite field is fixed).

Example 6 *Consider the mapping F over $\mathbb{F}_{2^n}$, for n odd, given as a polynomial $F(x) = x^3$, thus $\mathbb{F}_{2^n} \ni x \mapsto x^3 \in \mathbb{F}_{2^n}$. Since* $\gcd(3, 2^n) = 1$ *for odd k, F is a permutation. Furthermore, $N_F = 2^{n-1} - 2^{\frac{n-1}{2}}$ which is exceptionally high nonlinearity and such functions are called almost bent (AB) for this reason. The mapping x^{2^k+1} is also known as Gold mapping, when* $\gcd(k, n) = 1$.

Another important property of substitution boxes is their differential table. Actually, this property of having low uniformity of differentials is of the same importance as nonlinearity in the design of S-boxes since it leads to differential cryptanalysis which is one of the most powerful cryptanalytic tools.

Definition 9 *Let F be an (n, m) S-box, that is $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$. For any $a \in \mathbb{F}^n$ and $b \in \mathbb{F}^m$, we denote*

$$\delta_F(a, b) = \#\{x \in \mathbb{F}^n,\ F(X_n + a) + F(X_n) = b\} \tag{13}$$

where $\#S$ is the cardinality of any set S. We define

$$\delta(F) = \max_{a \neq 0, b \in \mathbb{F}^n} \delta_F(a, b). \tag{14}$$

The smaller the $\delta(F)$, the better the differential properties of F.

The above definition is more generally stated in terms of vector space mappings, since when $m \nmid n$ where is no corresponding finite field representation. In the Boolean case, when $m = 1$, the above differentials are commonly denoted as $D_{a,f}(x) = f(x + a) + f(x)$, which is a derivative of f in direction $a \neq 0$, and obviously $D_{a,f}(x) \in \mathcal{B}_n$.

Exercise 4 *Show that if* $\deg(f) = d$ *then* $\deg(D_{a,f}) \leqslant d - 1$.

Referring back to our finite filed representation we now assume that $n = m$ and consider the derivative of $F(x) \in \mathbb{F}_{2^n}[x]$ (the ring of polynomials with coefficients in $\mathbb{F}_2^n$). That is, for $F(x) \in \mathbb{F}_{2^n}[x]$ we consider the number of solutions to $F(x + a) + F(x) = b$, where $a \in \mathbb{F}_{2^n}^*$ and $b \in \mathbb{F}_{2^n}$. Notice that if x_0 is a solution to this equation for some fixed a and b then $x_0 + a$ is a solution as well. Also, if a is fixed then clearly $\sum_{b \in \mathbb{F}_{2^n}} \delta_F(a, b) = 2^n$. Therefore, the functions for which $\delta(F) = 2$ attains the lowest possible differential spectra and are called almost perfect nonlinear (APN) functions.

Remark 1 *The term perfect nonlinear functions is reserved for polynomials over $\mathbb{F}_q$ where the prime characteristic of the filed $p \neq 2$. In this case, there exists mappings $F(x) \in \mathbb{F}_q[x]$ such that $F(x+a) - F(x)$ is a permutation over F_q for any $a \in \mathbb{F}_q^*$, thus $F(x+a) - F(x) = b$ has exactly one solution for any $a \in \mathbb{F}_q^*$ and $b \in \mathbb{F}_q$. Such mappings are called* planar mappings *and the known classes mainly come from linearized polynomials. For instance, the mapping $F(x) = x^2$ is planar over $\mathbb{F}_{p^n}$, for $p \neq 2$, since $F(x+a) - F(x) = x^2 + 2ax + a^2 - x^2 = 2ax + a^2$ due to the fact that $\alpha x + \beta$ is a permutation over $\mathbb{F}_{p^n}$ for any nonzero α and any β.*

Example 7 *Let $F(x) = x^3$ over $\mathbb{F}_{2^n}$, where n is odd. Then, F is an APN permutation. The permutation property being clear, we need to show that $F(x+a) + F(x) = b$ admits at most two solutions for any $a \in \mathbb{F}_{2^n}^*$ and $b \in \mathbb{F}_{2^n}$. Indeed, $F(x+a) + F(x) = (x+a)^3 + x^3 = ax^2 + a^2x + a^3$ so that $ax^2 + a^2x + a^3 = b$ is of degree 2 and can have at most two solutions in the field.*

Since for any $\alpha \in \mathbb{F}_{2^n}$ we have $\alpha^{2^n-1} = 1$ it is sufficient to consider polynomials of degree up to $2^n - 1$, that is the polynomials of the form $F(x) = \sum_{i=0}^{2^n-1} a_i x^i$, where $a_i \in \mathbb{F}_{2^n}$. Notice that this global degree of a polynomial in $\mathbb{F}_{2^n}[x]$ does not correspond to the algebraic degree of F defined previously. More precisely, the algebraic degree of F corresponds to the largest Hamming weight of i for which $a_i \neq 0$, see Carlet [1] which is an excellent reference for all topics treated here. To realize this consider $F(x) = x^4$ over $\mathbb{F}_{2^n}$ whose algebraic degree is only 1 since it belongs to the class of linearized polynomials over the finite filed of the form $L(x) = \sum_{i=0}^{n-1} a_i x^{2^i}$. If $\alpha_1, \ldots, \alpha_n$ is a basis of $\mathbb{F}_{2^n}$ (through the isomorphism of $\mathbb{F}_2^n$ and $\mathbb{F}_{2^n}$) so that any element $x \in \mathbb{F}_{2^n}$ can be uniquely represented as $x = x_1\alpha_1 + \ldots + x_n\alpha_n$, where $x_i \in \mathbb{F}_2$, then,

$$x^4 = (x_1\alpha_1 + \ldots + x_n\alpha_n)^4 = x_1^4\alpha_1^4 + \ldots + x_n^4\alpha_n^4 = x_1\alpha_1^4 + \ldots + x_n\alpha_n^4,$$

since in the Boolean ring $x_i^2 = x_i$. In this representation we actually consider $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$, where $x = (x_1, \ldots, x_n) \mapsto (f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n))$, and each f_i is a linear Boolean function. Notice that $\alpha_1^4, \ldots, \alpha_n^4$ is just a linear transformation of the basis (Forbenius automormhism).

Example 8 *Let $F(x) = x^3$ over $\mathbb{F}_{2^3}$ defined by a primitive polynomial $p(x) = x^3 + x + 1$ over $\mathbb{F}_2$. Let α be primitive element of $\mathbb{F}_{2^3}$, i.e., $\alpha^3 = \alpha + 1$ and let $\{1, \alpha, \alpha^2\}$ be a polynomial basis of $\mathbb{F}_{2^3}$. Then the component functions of $F(x) = 1 \cdot f_1(x_1, x_2, x_3) + \alpha f_2(x_1, x_2, x_3) + \alpha^2 f_3(x_1, x_2, x_3)$ are derived as,*

$$\begin{aligned} F(x) &= x^3 = (x_0 + \alpha x_1 + \alpha^2 x_2)^3 = \\ &= (x_0 + \alpha x_1 + \alpha^2 x_2)(x_0 + \alpha x_1 + \alpha^2 x_2)^2 = \\ &= (x_0 + \alpha x_1 + \alpha^2 x_2)(x_0 + \alpha^2 x_1 + \alpha^4 x_2) \overset{\alpha^3=\alpha+1}{=} \ldots \\ &= (x_0 + x_1 + x_2 + x_1x_2) + \alpha(x_1 + x_0x_1 + x_0x_2) + \alpha^2(x_2 + x_0x_1) \end{aligned}$$

Notice that the algebraic degree of F above is 2 since the binary (Hamming) weight of 3 is $wt(3) = 2$. Concludingly, even though x^3 is an APN permutation and an AB function as well (thus achieving the maximum nonlinearity) its algebraic degree is low and therefore its use in block ciphers is not recommended. We conclude this section with one of the most elegant problem in the theory of finite fields (related to cryptography) which is the existence of APN permutations for even n.

Open Problem 3 *For even $n > 6$, find a class (or single function) which is an APN permutation or disprove their existence !! Only recently, Dillon [2] exceptionally confirmed the existence of such mappings for $n = 6$ using very sophisticated connections with coding theory.*

Vectorial bent functions

While the construction of Boolean bent functions (at least those in $\mathcal{M}$ class was easy and generic, the construction of $F : \mathbb{F}_2^n \to \mathbb{F}_2^k$ is not that obvious. Now we have to ensure that for $F(x) = (f_1(x), \ldots, f_k(x))$ all nonzero linear combinations of the form $a_1 f_1(x) + \ldots + a_k f_k(x)$ are bent, where f_i are Boolean functions. The bound on k for which it is possible to find such a collection was given by Nyberg [8], that is, $k \leqslant n/2$. The design of such functions achieving the upper bound on k, that is $k = n/2$, was only given in terms of sequences and the representation of these functions in [14] is not univariate (meaning that their representation as polynomials over finite fields is unclear). In a recent work [7], the structure of the cyclic group of the $2^k + 1$ roots of the unity was used to derive one complete class of vectorial bent functions $F : \mathbb{F}_2^n \to \mathbb{F}_2^{n/2}$ in a univariate representation.

Let us define the trace function $Tr_m^n : \mathbb{F}_{2^n} \to \mathbb{F}_{2^m}$, a mapping to a subfield $\mathbb{F}_{2^m}$ when $m \mid n$, is defined as

$$Tr_m^n(x) = x + x^{2^m} + x^{2^{2m}} + \ldots + x^{2^{(n/m-1)m}}, \text{ for all } x \in \mathbb{F}_{2^n}. \tag{15}$$

The absolute trace $Tr_1^n : \mathbb{F}_{2^n} \to \mathbb{F}_2$, also denoted by Tr, then maps to the prime field. Let also $n = 2k$, and denote by L the field $\mathbb{F}_{2^n}$ and its subfield $\mathbb{F}_{2^k}$ by K. Let $\mathcal{U} = \{u \in L : u^{2^k+1} = 1\}$ be the cyclic subgroup of L of order $2^k + 1$, which is essentially the group of $(2^k + 1)$th primitive roots of unity. Then, $\alpha^{2^k-1} = \beta$ is a generator of $\mathcal{U}$, and $\mathcal{U} = \{\alpha^{s(2^k-1)}, s = 0, \ldots, 2^k\}$, where $\alpha \in L$ is a primitive element. Now, any element $x \in L^*$ can be uniquely represented as $x = \gamma u$, where $\gamma \in K^*$ and $u \in \mathcal{U}$, and furthermore $\cup_{u\in\mathcal{U}}\, uK^* = L^*$. For convenience, we denote $P(x) = \sum_{i=1}^t a_i x^{i(2^k-1)}$ so that $F(x) = Tr_k^n(P(x))$. The following result specify three equivalent necessary and sufficient conditions (we only state two here) for F to be vectorial bent [7].

Theorem 1 *Let $n = 2k$, and define $F(x) = Tr_k^n(P(x))$, where $P(x) = \sum_{i=1}^t a_i x^{i(2^k-1)}$ and $t \leqslant 2^k$. Then the following conditions are equivalent:*

1. *F is a vectorial bent function of dimension k.*
2. *$\sum_{u\in\mathcal{U}} (-1)^{Tr_1^k(\lambda F(u))} = 1$ for all $\lambda \in K^*$.*
3. *There are two values $u \in \mathcal{U}$ such that $F(u) = 0$, and furthermore if $F(u_0) = 0$, then F is one-to-one and onto from $\mathcal{U}_0 = \mathcal{U} \setminus u_0$ to K.*

The proof is rather tedious but relies on the nice property of the exponents (known as Dillon exponent) of the terms $x^{i(2^k-1)}$. Indeed, since $x \in GF(2^n)$ can be written as $x = uy$ for $u \in U$, $y \in GF(2^k)$, then $F(uy) = \sum_{i=1}^t a_i (uy)^{i(2^k-1)} = \sum_{i=1}^t a_i u^{i(2^k-1)} = F(u)$, as $y^{i(2^k-1)} = 1$ for any y because $y \in K^*$. This means that F is constant on any coset uK^* which makes the analysis much easier.

Exercise 5 *(Semi-hard) Show the item (2) above by using the fact that F is vectorial bent if and only if $W_F(\lambda, \sigma) = \pm 2^k$ for any $\lambda \in K^*$ and any $\sigma \in L$. Here, $W_F(\lambda, \sigma) = \sum_{x\in L} (-1)^{Tr_1^k(\lambda F(x)) + Tr_1^k(\sigma x)}$. Use the representation $x = u\gamma$ for the elements in L^*, and that $F(u\gamma) = F(u)$ for any $\gamma \in K^*$. Thus $W_F(\lambda, \sigma)$ can be therefore written (using $F(0) = 0$) as $1 + \sum_{u\in U} \sum_{\gamma\in K^*} (-1)^{Tr_1^k(\lambda F(u\gamma)) + Tr_1^k(\sigma u\gamma)}$ $\ldots$*

We conclude this part by mentioning that there exist various generalizations of the concept of bent functions, for instance one may naturally define $F : \mathbb{F}_p^n \to \mathbb{F}_p^n$, for prime $p \neq 2$, but this requires a modification of the main cryptographic notions.

Graph theoretic aspects of Boolean functions

Let G be a multiplicative group of order v. A k-subset D of G is a (v, k, λ, μ) partial difference set (PDS) if each non-identity element in D can be represented as gh^{-1} $(g, h \in D, g \neq h)$ in exactly λ ways, and each non-identity element in $G \setminus D$ can be represented as gh^{-1} $(g, h \in D, g \neq h)$ in exactly μ ways. We shall always assume that the identity element 1_G of G is not contained in D. Using the language of group ring algebra $R[G]$, a k-subset D of G with $1_G \notin D$ is a (v, k, λ, μ)-PDS if and only if the following equation holds:

$$DD^{(-1)} = (k - \mu)1_G + (\lambda - \mu)D + \mu G, \tag{16}$$

where in $R[G]$ we denote $D = \sum_{g \in G} d_g g$ and $D^{(t)} = \sum_{g \in G} d_g g^t$, for $d_g \in R$. Combinatorial objects associated with partial difference sets are strongly regular graphs. A graph Γ with v vertices is called a (v, k, λ, μ) strongly regular graph (SRG) if each vertex is adjacent to exactly k other vertices, any two adjacent vertices have exactly λ common neighbours, and any two non-adjacent vertices have exactly μ common neighbours. Given a group G of order v and a k-subset D of G with $1_G \notin D$ and $D^{-1} = D$, the graph $\Gamma = (V, E)$ is called the Cayley graph generated by D in G and is defined as follows:

1. The vertex set V is G;

2. Two vertices g, h are joined by an edge if and only if $gh^{-1} \in D$.

The following result links together the notions of partial difference set and the property of a graph being strongly regular.

Theorem 2 (13) *) Let Γ be the Cayley graph generated by a k-subset D of a multiplicative group G with order v. Then Γ is a (v, k, λ, μ) strongly regular graph if and only if D is a (v, k, λ, μ)-PDS with $1_G \notin D$ and $D^{-1} = D$.*

Note that in the binary case, when Boolean functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ are considered, the Cayley graph is induced with respect to a subset of the elementary additivie Abelian 2-group $\mathbb{F}_2^n$. Since the condition that for $d \in D$ we must have $-d \in D$, any $D \subseteq \mathbb{F}_{2^n}$ will define the Cayley graph (each element is its own additive inverse) so that there is an edge between g and h if and only if $h \oplus g \in D$. The Cayley graph $\Gamma_f = (\mathbb{F}_2^n, E_f)$ associated to a Boolean function f is defined by selecting $D = \{x \in \mathbb{F}_2^n : f(x) = 1\}$ (D is called the support set of f) and defining the set of edges as,

$$E_f = \{(u, w) \in \mathbb{F}_2^n \times \mathbb{F}_2^n \mid f(\mathbf{u} \oplus \mathbf{w}) = 1\},$$

where for convenience we use the boldface to denote the elements of $\mathbb{F}_2^n$ so that $\mathbf{u} = (u_1, \ldots, u_n)$. The operation $\oplus$ over $\mathbb{F}_2^n$ is of course the componentwise modulo 2 addition. Furthermore, we specify the elements of $\mathbb{F}_2^n$ by using the decimal representation of their indices, thus $\mathbf{u}_0 = (0, \ldots, 0)$, $\mathbf{u}_1 = (1, \ldots, 0)$, $\ldots$, $\mathbf{u}_{2^n-1} = (1, \ldots, 1)$.

A graph is called *regular* of degree (valency) r if every vertex has degree (valency) r, that is, the number of edges incident to it is r. The Cayley graph Γ_f associated to any Boolean function f is obviously D regular. On the other hand, such a graph with parameters $(\mathbb{F}_2^n, D, d, e)$ is called *strongly regular graph* (SRG) if there exist nonnegative integers e, d such that for all vertices u, v the number of vertices adjacent to both u and v is e if u, v are adjacent, respectively, this number is d if u, v are nonadjacent. An easy counting argument shows that $D(D - d - 1) = e(v - D - 1)$. Notice that in general strongly regular graphs appear to be difficult to investigate.

The adjacency matrix A_f of size $2^n \times 2^n$ is the matrix whose entries are $A_{i,j} = f(\mathbf{u}_i \oplus \mathbf{u}_j)$, thus $A_{i,j} = 1$ if and only if $\mathbf{u}_i$ and $\mathbf{u}_j$ are connected. Given a graph Γ_f and its adjacency matrix A_f the *spectrum* *Spec*(Γ_f) is the set of eigenvalues of A_f. The following result specifies the eigenvalues in terms of Walsh coefficients and vice versa.

Theorem 3 *Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$, and let λ_i, $0 \leqslant i \leqslant 2^n - 1$ be the eigenvalues of its associated graph Γ_f. Then $\lambda_i = W_f(\mathbf{b}_i)$, for any i.*

Proof. The eigenvectors of the Cayley graph Γ_f are the characters $Q_{\mathbf{w}}(x) = (-1)^{\mathbf{w}\cdot x}$ of $\mathbb{F}_2^n$ [**?**]. Moreover, the i-th eigenvalue of A_f, corresponding to the eigenvector $Q_{\mathbf{b}_i}$ is given by $\lambda_i = \sum_{x\in\mathbb{F}_2^n}(-1)^{\mathbf{b}_i\cdot x} f(x) = W_f(\mathbf{b}_i)$. □

It is known that a connected r-regular graph is strongly regular if and only if it has exactly three distinct eigenvalues $\lambda_0 = r$ (or $\lambda_0 = D$ in our notation) and λ_1, λ_2. Furthermore, we have the following $e = r + \lambda_1\lambda_2 + \lambda_1 + \lambda_2$ and $d = r + \lambda_1\lambda_2$. It can be shown that bent functions, thus n is even, are the only Boolean functions whose associated Cayley graph is a strongly regular graph with $e = d$. In particular, for bent functions we have $\lambda_2 = -\lambda_1 = 2^{n/2-1}$ and $\lambda_0 = D = 2^{n-1} \pm 2^{n/2-1}$.

Exercise 6 *For $n = 4$ verify that $f(x_1, \dots, x_4) = x_1x_2 + x_3x_4$ is a bent function. Compute the parameters $e = d$.*

An additional property of bent functions is related to the notion of the triangle-free property. In other words, a graph is triangle-free if there are no paths of the form $xyzx$, where the vertices x, y, z are distinct. It can be shown that if Γ_f is triangle-free then f cannot be bent. But this property cannot be used for distinguishing the bent property of Boolean functions since the converse is not true. That is, there are functions whose graphs contain (many) triangles but they are not bent.

References

[1] C. Carlet. *Boolean Functions for Cryptography and Error Correcting Codes.* Cambridge University Press, 2010.

[2] J. Dillon. APN polynomials: An update. In *Fq9, the 9th International Conference on Finite Fields and Applications*, 2009.

[3] J. F. Dillon. Elementary Haddamard Difference Sets. Ph. D. thesis, University of Maryland, U.S.A., 1974.

[4] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes.* North-Holland, Amsterdam, 1977.

[5] J. L. Massey. Shift-register synthesis and BCH decoding. *IEEE Trans. on Inform. Theory*, IT-15(1):122–127, 1969.

[6] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, 1997.

[7] A. Muratovic-Ribic, E. Pasalic, and S. Bajric. An analysis of multiple output trace bent functions with nonlinear Niho exponents using symmetric polynomials. *IEEE Trans. on Inform. Theory*, IT-60(2):1337–1347, 2014.

[8] K. Nyberg. Perfect nonlinear S-boxes. In *Advances in Cryptology—EUROCRYPT'91*, volume LNCS 547, pages 378–385. Springer-Verlag, 1991.

[9] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, Vol. 27:379–423 (Part I) and 623–656 (Part II), 1948.

[10] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, Vol. 27:656–715, 1949.

[11] T. Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Trans. on Inform. Theory*, IT-30:pages 776–780, 1984.

[12] G. J. Simmons. *Contemporary Cryptology.* Wiley-IEEE Press, New York, 1999.

[13] G-Z. Xiao and J. L. Massey. A spectral characterization of correlation-immune combining functions. *IEEE Trans. on Inform. Theory*, IT-34:569–571, 1988.

[14] A. M. Youssef and G.. Gong. Hyper-bent functions. In *Advances in Cryptology—EUROCRYPT 2001*, volume LNCS 2045, pages 406–419. Springer-Verlag, 2001.