Chapter 13

WALSH-FOURIER ANALYSIS OF BOOLEAN COMBINERS IN CRYPTOGRAPHY

Franz Pichler
Professor Emeritus, Johannes Kepler University Linz,
A-4040 Linz, Austria
franz.pichler@jku.at

Abstract The paper presents a brief overview of applications of Walsh functions in cryptography.

1. Introduction

The theory of Walsh functions goes back to the original paper of Walsh (1923). This was followed by contributions of Paley, Fine and others in pure mathematics. After WWII interest in *communication engineering* and *signal processing* arose and research, mainly in the USA-there at Jet Propulsion Laboratory in Pasadena but also at companies and universities was done. From 1970 on regularly conferences first at the *Naval Research Laboratory* in Washington D.C., were started, with *H. Harmuth* (see, for example, [3], [4]) as its main speaker. It is most likely that at this time the importance of Walsh functions for the characterization of Boolean function as they are applied in cryptography was already known to different organizations but was kept confidential. [1]

2. Walsh Functions: General overview on the theory

Walsh functions have become important for the analysis of Boolean functions by their application in cryptography in combiners and also in S-boxes. Their mathematical theory is highly developed. They are *character functions* of a specific abelian group, *the dyadic group* and the related theory is a special

[1] This paper is a part of a lecture series of the author on Cryptology and is addressed to Non-specialists in the field of Walsh functions. Additional citations may be found in the LNCS publications of the EUROCRYPT and CRYPTO Conferences.

case of the theory characters and of the field of *abstract harmonic analysis* (see for example the book of Rudin [8] or Hewitt-Ross [5]). The case we are dealing with is given by the finite dyadic group $D(n)$ which is the n-fold direct product of the cyclic group Z_2. In this case the theory becomes a part of *linear algebra*.

The *dyadic group* $D(n)$ is defined by $D(n) := (B^n, \oplus)$, its elements $x = (x_0, x_1, \ldots, x_{n-1})$ are Boolean n-tuples, the addition $x \oplus y$ of the elements x and y is coordinate-wise done.

A *Boolean function* f is defined by a function f from B^n to B.

Let $Z(n)$ denote the set $Z(n) := \{0, 1, \ldots, 2^n - 1\}$. There exists a one to one correspondence between $Z(n)$ and $D(n)$ by the function bin with

$$bin(x_0 + 2x_1 + \ldots + 2^{n-1}x_{n-1}) := (x_0, x_1, \ldots, x_{n-1}).$$

For elements t from $Z(n)$ we use often the notation $t = (t_0, t_1, \ldots, t_{n-1})$ and extend the xor operation $\oplus$ also to $Z(n)$.

Walsh functions $w(s, \cdot)$ are usually defined as real-valued functions $w(s, \cdot) : Z(n) \to R$ by

$$w(s, t) = (-1)^{<s,t>}, \quad s \in Z(n),$$

where $< s, t >$ denotes the inner product $s_0t_0 + s_1t_1 + \cdots + s_{n-1}t_{n-1}$ of s with t.

Walsh functions $w(s, \cdot)$ take only values $+1$ and -1. The *Walsh transform* $\hat{F}$ of a function $F : Z(n) \to R$ is defined by

$$\hat{F}(s) := \sum_t F(t)w(s, t).$$

The *inverse Walsh transform* of $\hat{F}$ is given by

$$f(t) = \frac{1}{2^n} \sum_s \hat{F}(s)w(s, t).$$

Let F, G denote functions from $Z(n)$ to R. *The dyadic convolution product* $F * G$ is defined as the function

$$(F * G)(t) = \sum_a F(t \oplus a)G(a).$$

For dyadic convolution the following theorem called *the dyadic convolution theorem* is valid

$$\widehat{(F * G)} = \hat{F} \cdot \hat{G}.$$

Notice that for $F * G = E$ (E the function $E(0) = 1$, $E(t) = 0$, else) it follows $F = E$.

Notice that for $F * G = E$ (E the function $E(0) = 1$, $E(t) = 0$ else) it follows $F = E$.

Let F_a denote the a-dyadic shifted function $F_a(t) := F(t \oplus a)$. The following *dyadic shifting theorem* is easy to prove

$$\hat{F}_a = w(a, \cdot)\hat{F}.$$

From the formula for the Walsh transform of a function F, we get for $\hat{F}(0)$ the value

$$\hat{F}(0) = \sum_t F(t),$$

and from the formula for the inverse Walsh transform

$$F(0) = \frac{1}{2^n} \sum_s \hat{F}(s).$$

The dyadic cross correlation function $DCC(F, G)$ of functions F and G is defined by

$$DCC(F, G) = \sum_t F(t \oplus a)G(t).$$

The dyadic autocorrelation function $DAC(F)$ of a function F is defined by

$$DAC(F)(a) = \sum_t F(t \oplus a)F(t).$$

For the DAC of a function F the following theorem can be proven

$$\widehat{DAC}(F) = F^2,$$

which is a mathematical expression of *the Theorem of Wiener-Khintchin*.

Of specific interest are functions F which take (as the Walsh functions) only values $+1$ and -1 on $Z(n)$. The following theorem characterizes such functions by spectral properties.

THEOREM 13.1 *A function F is a "$+1/-1$ function" if and only if the following equation is valid*

$$\hat{F} * \hat{F} = 2^n E.$$

An interesting theorem is the following.

THEOREM 13.2 *(Theorem of Liedl)*
Let F be a polynomial of degree $m < n$. Then, $\hat{F}(s) = 0$ for all s with $\|s\|_H > m$, where $\|s\|_H$ denotes the Hamming weight *of s.*

3. Walsh Fourier Analysis of Boolean Functions

Applications of the theory of Walsh functions in the field of *cryptology* deal mainly with Boolean functions $f : B^n \rightarrow B$. Any such function f has a corresponding $+1/-1$ function F which is given by $F(t) = (-1)^{f(x)}$ where $x = bin(t)$. In the following, F has always this meaning.

The Walsh transform $\hat{f}$ of a Boolean function f is defined in the following way

$$\hat{f}(y) = \sum_x (-1)^{f(x)} (-1)^{<x,y>},$$

or since $f(x) + < y, x > (\text{mod } 2) = f(x) \oplus < y, x >$ we have also

$$\hat{f}(y) = \sum_x (-1)^{f(x) \oplus <x,y>}.$$

It is to observe that the Walsh transform $\hat{f}$ of a Boolean function f is real-valued.

The dyadic autocorrelation and the dyadic cross correlation of a Boolean function f is defined by the DAC and DCC of the associated $+1/-1$ function F:

$$DAC(f) := DAC(F)$$

and

$$DCC(f) := DCC(F).$$

The following results for Boolean functions f are valid:

$$DAC(f)(0) = 2^n,$$

and

$$\|f \oplus f_a\|_H = 1/2 - 1/2^{n+1} DAC(f)(a).$$

It can be observed that for $a = 0$ as expected $\|f \oplus f\|_H = 0$. Furthermore that the Hamming distance of f and f_a for $a \neq 0$ is close to $\frac{1}{2}$ if $DAC(f)(a)$ is small. This is the case for the Boolean functions $f : D(2^n - 1) \rightarrow B$ which are generated by a maximum length linear feedback shift register MLFSR of length n (pseudo random code-words).

A main application of the Walsh transform in cryptology is given by the spectral characterization of Boolean functions.

A Boolean function f on $D(n)$ is called *balanced* if

$$card\{x : f(x) = 0\} = card\{x : f(x) = 1\} = 2^n/2.$$

We have the "theorem": A function f is balanced if $\hat{f}(0) = 2^n/2$.

A Boolean function f satisfies by definition *the propagation criteria with respect to* $a \in D(n)$ if $f \oplus f_a$ is balanced. Here f_a denotes the dyadic a-shift of f which is given by $f_a(x) := f(x \oplus a)$.

A Boolean function f satisfies by definition the propagation criteria of degree k if it satisfies the propagation criteria for all $a \in D(n)$ with $0 < \|a\|_H = k$. In the case $k = 1$ we say that f satisfies *the Strict Avalanche Criteria* (SAC). The following theorem can be proven for the SAC of a Boolean function f:

THEOREM 13.3 *([1]*
A Boolean function f satisfies the Strict Avalanche Criteria if

$$\sum_s (\hat{f})^2(s)(-1)^{s_i} = 0,$$

for all i with $1 \leq i \leq n$.

The distance $d(f, g)$ between two Boolean functions f and g is given by

$$d(f, g) = \|f \oplus g\|_H.$$

Linear Boolean functions $l(y)$ are of the form $l(y)(x) =< y, x >$ or $l(y) = 1 \oplus < y, x >$.

A degree of non-linearity of a Boolean function f can be measured by its distance to a linear Boolean function. The following theorem allows to express the distance of a Boolean function f to the linear functions $l(y)$ by means of its spectrum:

THEOREM 13.4 *For a Boolean function f and a linear Boolean function $l(y)$*

$$d(f, l(y)) = \frac{1}{2}(2^n \hat{f}(y)).$$

In stream cipher architectures the analysis of the Boolean function which realizes a static combiner is of specific importance. To block correlation attacks to investigate the used secret key a sufficient degree m of correlation immunity of such a function is required. In this respect the following definition is introduced:

A Boolean function f is called to be *correlation immune* of order m if $f(x_1, x_2, \ldots, x_n)$ is statistically independent from every k-tupel, where $k < m + 1$, when considered as independent uniformly distributed binary random variables of stochastic processes $X_{i_1}, X_{i_2}, \ldots, X_{i_n}$.

For the characterization of a Boolean function with respect to its correlation immunity the following theorem is of importance.

THEOREM 13.5 *([10])*
A Boolean function f is correlation immune of order m if and only if $\hat{f}(y) = 0$ for all y with $\|y\|_H \leq m$, where $\|y\|_H$ denotes the Hamming weight of y.

In the terminology of Walsh-Fourier analysis this means, that a Boolean function f which is correlation immune of order m contains in its Walsh-Fourier representation only Walsh functions, which are a product of more than m *Rademacher functions*. The related Walsh-Fourier spectrum $\hat{f}$ can therefore be considered as a nonlinear function which compares to polynomials of higher degree than m.

4. Design of Boolean Function Combiners

The determination of a Boolean function which meets the necessary requirements is an important mathematical task in the cryptography of stream ciphers. We explore in detail the following properties, which have some relevance.

If $x_1, x_2, x_3, \ldots, x_n$ denotes the pseudo random streams received by the combiner C, then the resulting output stream y of the considered combiner $C(x_1, x_2, x_3, \ldots, x_n)$ should be "cryptologically improved" compared to the individual input streams x_i $(i = 1, 2, 3, \ldots, n)$.

A combiner C must not "leak" (should have a strong one way property to make cryptanaysis difficult).

The design of combiners for strong pseudo random generators used in cryptography is usually a part of a trade secret of companies. However there are a number of published results which can give an orientation. Most of publications deal with static combiners, based on Boolean functions, only a few results are known for dynamic combiner.

A Boolean combiner can be realized by a properly chosen Boolean function C from B^n to $B := \{0, 1\}$. A Boolean combiner can be represented either by a table or by a Boolean expression. Usually it is to assume that C is given by its *Algebraic Normal Form* $ANF(C)$.

One of the most important requirements in the design of Boolean combiners concerns the degree of correlation immunity $I(C)$ to avoid leaking with respect to the correlation attack (described by Siegenthaler [9] and Golic [2]) $I(C)$ can be determined by spectral properties of the Walsh-Fourier transform $WFT(C)$ of C. A sufficient degree $I(C)$ needs a certain degree of nonlinearity of the discrete polynomial associated to C. The results, which are already described in Section 2 were derived by the work of Xiao and Massey [10].

It is possible to construct a sufficiently large number of correlation immune Boolean functions for any desired degree m [6]. Other results which are derived by Siegenthaler [9] are based on repeated algebraic computations.

If the required degree $I(C)$ of correlation immunity of a combiner C is given, then the following recipe for the construction of a combiner C with degree $I(C) = m$ can be applied:

1 Define C by $C(x_1, x_2, \ldots, x_n) := x_1 \oplus x_2 \oplus \cdots x_m \oplus g(x_{m+1}, \ldots, x_n)$ with a Boolean function $g : B^{n-m} \rightarrow B$.

2 Chose g such that the additional required properties of C are fulfilled.

In the following we explore some additional features and their spectral representation by Walsh-Fourier representations which are used in combiner design. In doing this we have to distinguish between *static combiners* represented by Boolean functions (Boolean function combiner) and *dynamic combiners* (FSM combiners, also called in cryptography "combiners with memory") represented by *finite state machines*.

In both cases it is the goal to derive from observed output bits

$$y(0)y(1)y(2), \ldots, y(k), \ldots$$

of the combiner C some knowledge about the input streams $x_1, x_2, \ldots, x_n$ and the "machines" (specifically their initial states) $M_1, M_2, \ldots, M_n$ which generate it. To get such a useful knowledge for the mounting of an attack this should be computational hard. In the case of dynamic combiners our consideration is restricted to finite state machines which are finite memory machines. In this case the problem of the design of a dynamic combiner can be reduced to the design of a Boolean function combiner.

Boolean function combiners are designed by switching functions f such that

1 The solution of the system of equations $f(x(i)) = y(i), i = 0, 1, 2, \ldots, k$; $x(i) = (x_1(i), x_2(i), \ldots, x_n(i))$ is computational hard.

2 A correlation analysis between the output stream y and the individual input streams x_i $(i = 1, 2, \ldots, n)$ shows no results regardless of the length of the applied streams y and x_i.

The condition (1) requires the highly nonlinear functions f. In the condition (2), in contradiction, however, certain linear component of f is required to meet correlation immunity requirements.

Different ways to represent switching functions f are known:

1 By *the disjunctive form* DF,

2 By *the conjunctive form* CF,

3 By *the algebraic normal form* ANF (a *multivariate polynomial*).

In cryptanalysis it is for algebraic reasons often desirable to use the ANF of a Boolean function f which is given by

$$\begin{aligned} ANF(f) \quad := \quad & a_1x_1 + a_2x_2 + \cdots + a_nx_n + a_{1,2}x_1x_2 + a_{1,3}x_1x_3 \\ & + \cdots + a_{n-1,n}x_{n-1}x_n + a_{1,2,3}x_1x_2x_3 + a_{1,2,4}x_1x_2x_4 \\ & + \cdots + a_{1,2,3,4,\ldots,n}x_1x_2\cdots x_n, \end{aligned}$$

there exist methods to compute DF, CF and ANF from any each other.

In Cryptology the following criteria are considered as useful in the design and analysis of Boolean combiners and Boolean networks ("S-boxes")

1 Balance,

2 Nonlinear order,

3 Correlation immunity,

4 Bentness,

5 Distance to linear structures,

6 Strict avalanche criterion,

7 Propagation characteristic,

8 Global avalanche criterion.

The criteria (1)-(8) can be defined as shown in the following. Also their characterization, if possible, in the spectral domain is given:

A Boolean function f is called *balanced* if

$$card\{x : f(x) = 1\} = card\{x : f(x) = 0\}.$$

The nonlinear order of f is defined by the maximal numbers of variables which appear in the $ANF(f)$.

A Boolean function f is *correlation-immune of order* m if the value of f is statistically independent from any m-tupel (compare with a more detailed definition in Section 2 of this paper).

These properties can be characterized by the Walsh-Fourier spectrum $\hat{f}$ of f in the following way:

THEOREM 13.6 *A Boolean function f is balanced iff $\hat{f}(0) = 0$.*

THEOREM 13.7 *A Boolean function f is correlation immune of order m iff $\hat{f}(w) = 0$ for all w with the Hamming weight $\|w\|_H \leq m$.*

THEOREM 13.8 *A Boolean function f is said to satisfy the strict avalanche criterion (SAC) if*

$$Pr\{f(x) \oplus f(x \oplus a) = 1\} = \frac{1}{2}$$

for $\|a\| = 1$.

THEOREM 13.9 *A Boolean function f satisfies the propagation characteristic (PC) of degree k if*

$$Pr\{f(x) \oplus f(x \oplus a) = 1\} = \frac{1}{2},$$

for $1 \leq \|a\| \leq k$.

REMARK 13.1 *The perfect nonlinearity requires a PC of degree n.*

THEOREM 13.10 *The global avalanche criterion GAC of a Boolean function f can be characterized by the dyadic autocorrelation function $DAC(f)$ of f which is given by*

$$DAC(f) := \sum_x F(x)F(x \oplus a).$$

A "good" GAC means that $DAC(f)$ is close to zero for almost all nonzero values of a and for $a = 0$ we should have $DAC(f)(0) = 2^n$. The Walsh-Fourier transform $WFT(DAC(f))$ is according to the Wiener-Khintchin theorem *the Walsh Power Spectrum* $P(f)$ of the Boolean function f. For functions f with good GAC the related $P(f)$ is almost constant (has a "white noise" characteristic).

Bent functions are Boolean functions f which satisfy the propagation characteristic PC by degree n. For bent functions the following theorem is valid.

THEOREM 13.11 *A Boolean function f is a bent function if the modulus of $\hat{f}$ ($\hat{f}$ the Walsh transform of f) is constant with $\hat{f}(w) = 2^n/2$ for all $w \in GF(2)^n$.*

Satisfying the criteria (1)-(8) may lead to conflicts. Such examples are as follows:

1 Usually it is required that C is balanced, so it cannot be a bent function.

2 Bent functions does not exist if n is odd.

3 High linear order means low degree of correlation immunity.

To avoid a trade-off of the kind (1) it can be suggested to use static combiners C of the form

$$C = x_1 \oplus x_2 \oplus x_3 \oplus \cdots \oplus x_m \oplus C'(x_{m+1}, x_{m+2}, \cdots, x_n),$$

which is of the correlation immunity of degree m.

To meet additional criteria the Boolean function C' has to be designed accordingly.

5. Finite Memory FSM Combiners

For special cases of finite state machines the design of dynamic combiners can be reduced to the design of a static (Boolean) combiner. One class is given by finite state machines which possess the "finite memory property". Such finite state machines are called *Finite Memory Machines* (FMM). This leads to the concept of a *Finite Memory FSM combiner* (FMM combiner).

In the following the main results for the design of FMM-combiners are stated, see [7].

DEFINITION 13.1 *A finite state machine FSM has a finite memory of degree i if any simple experiment (w, v) observed on the FSM determines uniquely the reached state q as soon as the length of the experiment is equal to i or is larger than i, length $(w, v) = i$ or length $(w, v) > i$.*

There exist efficient algorithms to determine if a given FSM is a finite memory machine FMM and also to determine the degree i. If a FSM is a FMM then there is the possibility to compute the associated canonical shift register representation, which contains two feed-forward registers for the input and the output together with the Boolean output function $f(FMM)$.

In a FMM state transition is reduced to simple shift operation.

The following procedure for constructing a FMM combiner is suggested:

1 Feed the input streams $x_1, x_2, \ldots, x_n$ to the associated feed-forward shift register $R_1, R_2, \ldots, R_n$ of lengths $m_1, m_2, \ldots, m_n$.

2 Feed the output stream y to a feed-forward shift register R of length m.

3 Take all register states as inputs of the output function $f(FMM)$.

The resulting FMM has a finite memory of degree

$$i = max(m_1, m_2, \ldots, m_n, m).$$

The following steps can use the methods for the cryptographic design of the Boolean function $f(FMM)$ with the cryptanalytic methods known for Boolean function combiners. To camouflage the design the change of the

state coordinates of the FMM is recommended to change by state assignment the states such that the FMM canonical form disappears so that, after implementation, it is not immediately recognized as a FMM combiner on hardware blueprints.

References

[1] Forre, R., "A fast correlation attack on nonlinearly feedforward filtered shift-register sequences", *Advances in Cryptology - EUROCRYPT '89*, Springer-Verlag, 1989, 586-595.

[2] Golic, J., "Intrinsic statistical weakness of keystream generators", *Advances in Cryptology - ASIACRYPT '94*, Springer-Verlag, 1994, 91-103.

[3] Harmuth, H.F., *Transmission of Information by Orthogonal Functions*, 2nd ed. Springer-Verlag, New York, 1972.

[4] Harmuth, H.F., *Sequency Theory, Foundations and Application*, Academic Press, New York, 1977.

[5] Hewitt, E., Ross, K., *Abstract Harmonic Analysis*, Berlin, Springer, 1963.

[6] Pichler, F., "On the Walsh Fourier analysis of correlation immune switching functions", *Eurocrypt 86*, Linköping, Sweden (published in LNCS proceedings).

[7] Franz Pichler, "Konstruktion korrelationsimmuner Schaltfunktionen und Schaltwerke mittels Walsh-Fourieranalyse", *Contributions to General Algebra*, 6, Teubner, Stuttgart, 1988 (in German).

[8] Rudin, W., *Fourier Analysis on Groups*, Interscience Publisher, New York, 1960.

[9] Siegenthaler, T., "Design of combiners to prevent divide and conquer attacks", *Advances in Cryptology - CRYPTO '85*, Springer-Verlag, 1985, 273-279.

[10] Xiao, G., Massey, J., "A spectral characterization of correlation-immune combining functions", *IEEE Trans. Inform. Theory*, Vol. 34, 1988, 569-571.