Towards a spectral approach for the design of self-synchronizing stream ciphers

Jeremy Parriaux, Philippe Guillot, Gilles Millérioux

Towards a Spectral Approach for the Design of Self-Synchronizing Stream Ciphers

Jérémy Parriaux[1], Philippe Guillot[2], and Gilles Millérioux[1]

[1] Nancy University, CNRS,
Research Center for Automatic Control of Nancy (CRAN UMR 7039), France,
jeremy.parriaux@esstin.uhp-nancy.fr,
gilles.millerioux@esstin.uhp-nancy.fr,
[2] University of Paris 8
Laboratoire Analyse, Géométrie et Applications (LAGA UMR 7539), France
philippe.guillot@univ-paris8.fr

Keywords: Self-synchronizing stream cipher, Boolean function, Walsh matrix, correlation matrix

Abstract. This paper addresses the problem of characterizing the functions that can be used in the design of self-synchronizing stream ciphers. We propose a general framework based on a spectral characterization through correlation matrices or equivalently through Walsh matrices. Two modes of self-synchronization are discussed: the finite time one and the statistical one.

1 Introduction

Stream ciphers are cryptosystems specifically devoted to the transmission of data streams over public channels. At the transmitter side, the ciphertext is carried out by adding a plaintext symbol with a symbol of a pseudorandom stream called the key-stream. At the receiver side, the decryption consists in subtracting the ciphertext symbol with a symbol of, again, a pseudorandom stream. Proper decryption is achieved provided that the pseudorandom streams generated at the transmitter and receiver sides are the same. In other words, the pseudorandom generators have to be synchronized. There are two ways to ensure the synchronization. The first one is to use an external protocol in order to initialize the two generators with the same seed. The protocol must also be able to resynchronize the generators if the synchronization is lost. The resulting ciphers are known as synchronous stream ciphers. The second method relies on systems for which synchronization is due to a structural property. The corresponding ciphers are called self-synchronizing stream ciphers, SSSC for short. The absence of synchronization protocol makes them particularly appealing when high throughputs are required. As it turns out, very few works have paid attention to them. Let us mention [1,2] for exceptions. This work aims at characterizing new functions which can be involved in self-synchronizing stream ciphers. The interest of enlarging the class of candidate functions lies in that they can potentially lead to

systems of reduced size or with better cryptographic properties than the existing ones. The characterization is performed in the spectral domain and thereby allows to connect the results to the usual cryptographic criteria.
The outline of this paper is the following: Section 2 is devoted to the problem statement. Section 3 recalls the usual material devoted to spectral analysis and Boolean functions. Section 4 deals with the spectral characterization of the self-synchronizing property and is the core of the paper. Section 5 investigates the reachability of the states in terms of probability law. Finally, Section 6 is devoted to an illustrative example.

2 Problem Statement

Let us first introduce the notations. The two-element field is denoted by $\mathbb{F}_2$. The plaintext symbol to be ciphered at time $t \in \mathbb{N}$ is $m_t \in \mathbb{F}_2$, the corresponding ciphertext is $c_t \in \mathbb{F}_2$ and the corresponding recovered plaintext is $\widehat{m}_t \in \mathbb{F}_2$. In stream ciphers, the ciphertext c_t is obtained from the plaintext m_t by adding a random symbol $z_t \in \mathbb{F}_2$. The original message $\widehat{m}_t$ is recovered by subtracting the symbol $\widehat{z}_t \in \mathbb{F}_2$ from the ciphertext c_t. For binary streams the subtraction is the same operation than the addition. In the canonical representation of a self-synchronizing stream cipher, the random symbols z_t and $\widehat{z}_t$ are generated using the same keyed function $g_\theta : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$ whose arguments are a finite sequence of some past ciphertexts, namely $c_{t-1}, \ldots, c_{t-n}$. The parameter θ is the key of the system. The decryption is properly performed, that is $\widehat{m}_t = m_t$, whenever $\widehat{z}_t = z_t$. It is guaranteed if both the encryptor and the decryptor have same key and if the ciphertexts $c_{t-1}, \ldots, c_{t-n}$ are properly transmitted. The equations of the canonical form of self-synchronizing stream ciphers are

$$\begin{cases} z_t = g_\theta(c_{t-1}, \ldots, c_{t-n}) \\ c_t = m_t + z_t \end{cases} \text{(encryptor)} \tag{1}$$

$$\begin{cases} \widehat{z}_t \ = g_\theta(c_{t-1}, \ldots, c_{t-n}) \\ \widehat{m}_t = c_t + \widehat{z}_t \end{cases} \text{(decryptor)} \tag{2}$$

Remark 1. As the synchronization, in general, operates on bit streams, it is relevant to consider binary input devices. Moreover, the generalization to device with input and output symbols of k bits is direct which allows for instance to apply the results to byte streams. The extension of the results to this case is discussed after the main results.

The canonical form admits an equivalent recursive form involving an internal state $x \in \mathbb{F}_2^n$ which is an n–dimensional Boolean vector. Its value at time t is $x_t = (c_{t-1}, \ldots, c_{t-n})$. Its i^{th} coordinate is denoted by $(x_t)_i$. The corresponding block diagram is depicted in Figure 1. The equations read

$$\begin{cases} (x_{t+1})_i = (x_t)_{i-1} \text{ if } i > 0, \ c_t \text{ if } i = 0 \\ z_t \quad\quad = g_\theta(x_t) \\ c_t \quad\quad = m_t + z_t \end{cases} \text{(encryptor)} \tag{3}$$

$$\begin{cases} (\widehat{x}_{t+1})_i = (\widehat{x}_t)_{i-1} \text{ if } i > 0,\ c_t \text{ if } i = 0 \\ \widehat{z}_t \quad\ = g_\theta(\widehat{x}_t) \qquad\qquad\qquad\qquad \text{(decryptor)} \\ \widehat{m}_t \quad = c_t + \widehat{z}_t \end{cases} \tag{4}$$

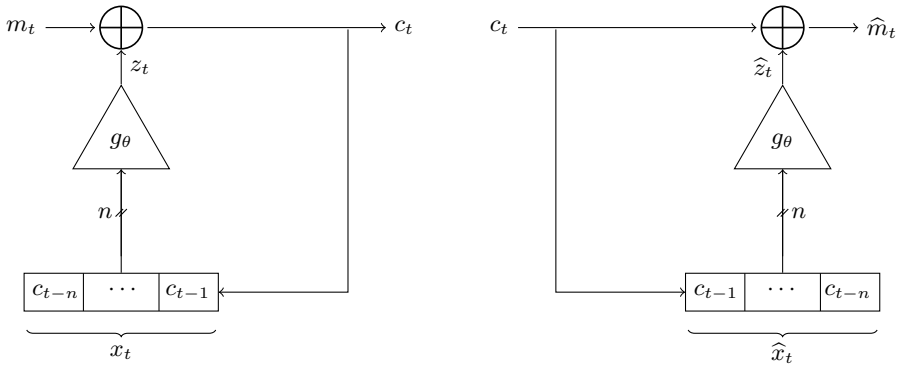

Fig. 1: Canonical recursive form of self-synchronizing stream ciphers

The canonical recursive form (3)–(4) is directly obtained from the canonical form (1)–(2). The state updating transformation is a mere shift register fed with the previous ciphertexts. Thus, the initial state is eliminated in a shift-like way and all the complexity of the system lies in the function g_θ. More interesting schemes are obtained when considering a keyed state updating transformation $f_\theta : \mathbb{F}_2 \times \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$ more complex than a shift. In this situation, the shift next-state function and the output function g_θ are replaced by a function f_θ and an output function $h_\theta : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$. This setup is referred to as the generalized recursive form and its block diagram is depicted in Figure 2. The corresponding equations are

$$\begin{cases} x_{t+1} = f_\theta(c_t, x_t) \\ z_t \quad = h_\theta(x_t) \quad \text{(encryptor)} \\ c_t \quad = m_t + z_t \end{cases} \tag{5}$$

$$\begin{cases} \widehat{x}_{t+1} = f_\theta(c_t, \widehat{x}_t) \\ \widehat{z}_t \quad = h_\theta(\widehat{x}_t) \quad \text{(decryptor)} \\ \widehat{m}_t \quad = c_t + \widehat{z}_t \end{cases} \tag{6}$$

In order to guarantee the self-synchronization property of the system, the function f_θ cannot be chosen arbitrarily. It must have the property that, after a fixed number of iterations, denoted by t_c, the key stream symbols z_t and $\widehat{z}_t$ are equal for all $t > t_c$. In the general case, this is achieved if and only if the current state of the decryptor is equal to the current state of the encryptor, $\widehat{x}_t = x_t$ regardless of the initial states x_0 and $\widehat{x}_0$. Clearly, given the system described by (5)–(6),

the self-synchronization can be studied by focusing exclusively on the function f_θ. Besides, the fact that this recursive form is more general than a mere shift allows to relax the constraint that the synchronization is achieved within a finite amount of time. That leads to so-called statistical self-synchronizing stream ciphers. They will be detailed and motivated later on in this paper.

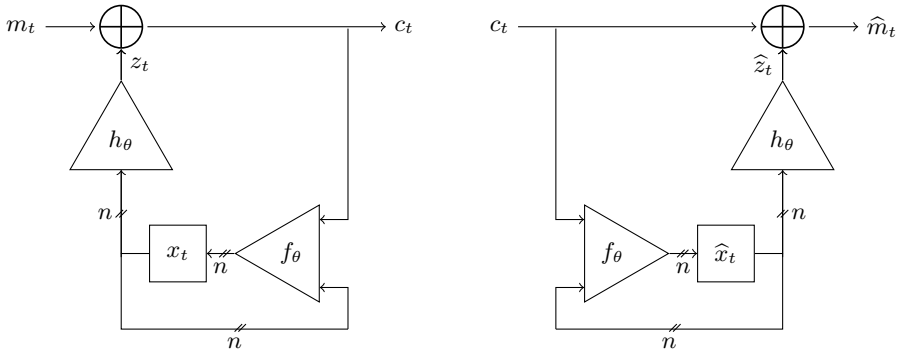

Fig. 2: Generalized recursive form of self-synchronizing stream ciphers

This paper does not intend to study how the key θ is involved in the system. Therefore, for simplification purposes and hereafter, the subscript θ will be omitted, the parametrization with the key of the functions will be implicit. The introduction of the key is left for a further study. In the sequel, a function from the vector space $\mathbb{F}_2^n$ to $\mathbb{F}_2$ will be referred to as a (n)–function. We will call a (n, m)–function a function from the vector space $\mathbb{F}_2^n$ to the vector space $\mathbb{F}_2^m$. Indeed, a (n, m)–function f is nothing but a m–dimensional vector whose each coordinate is a (n)–function. The j^{th} coordinate is denoted by f_j and named the coordinate function. We recall a definition introduced by Klimov and Shamir in [3].

Definition 1 (T–function). *A (n, n)–function is called a T–function if the coordinate function f_j depends only on the variables x_i with $i = 0, \ldots, j$.*

Some special T–functions of interest in our study are called strict T–functions, their definition is the following:

Definition 2 (Strict T–function). *A T–function such that the coordinate function f_j depends only on the variable x_i with $i = 0, \ldots, j-1$ is called a strict T–function.*

Note 1. What we call *strict T–function* is called *parameter* in [4] however, we think that this name is misleading in our context and we prefer to use another name in order not to confuse the reader.

Having a look at the literature, it can be noticed that, so far, all the self-synchronizing stream ciphers use the same principle in order to guarantee that the current state at time t does no longer depend on the initial state, that is to guarantee the self-synchronization. The state updating function is such that its coordinate functions depend on the bits of the internal state with strictly lower indexes than their own index. In other words, the state updating function is based on strict T–functions.

The main purpose of this paper is to pinpoint more general classes of functions which guarantee the self-synchronization property besides strict T–functions. Self-synchronization properties are addressed from a spectral point of view as motivated in the introduction.

3 Preliminaries

This section introduces a formal definition of self-synchronization and then recalls the strict necessary prerequisites on spectral analysis of Boolean functions from which our results will be derived.

3.1 Self-synchronization

Let us first formally define some self-synchronization related notions.

Definition 3 (Self-synchronizing sequence). *A ciphertext sequence (c) is self-synchronizing for f if there exists an integer t_c so that for all initial states x_0 and $\widehat{x}_0$*

$$\forall t \geq t_c,\ x_t = \widehat{x}_t \tag{7}$$

Definition 4 (Finite-time self-synchronization). *The system (5)–(6) is finite-time self-synchronizing if the minimum value t_c is upper bounded for all possible ciphertext sequences (c). The upper bound t_c is called the self-synchronization delay of f.*

Remark 2. Finite-time self-synchronization means that there is an integer t_c such that any sequence of length at least t_c is a self-synchronizing sequence. The synchronization delay depends on the pair of initial states x_0 and $\widehat{x}_0$. The delay t_c is defined as the maximum delay over all initial state pairs.

Definition 5 (Finite-time self-synchronizing function). *A $(n+1, n)$–function f is called finite-time self-synchronizing function if, when used as a next-state function in the system (5)–(6), the resulting system is finite-time self-synchronizing.*

3.2 Spectral Analysis

The rest of this section recalls the basics about Boolean spectral analysis. If f is a (n)–function, we denote by $\widehat{f}$ its Fourier transform, which is by definition the real-valued mapping $\mathbb{F}_2^n \longrightarrow \mathbb{R}$ defined for any $u \in \mathbb{F}_2^n$, by

$$\widehat{f}(u) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{x \cdot u} \tag{8}$$

where $x \cdot u = x_0 u_0 + \cdots + x_{n-1} u_{n-1}$.
The expression of the Walsh transform (8) also admits a matrix oriented representation $\widehat{f} = Hf$ where H is the so called Hadamard matrix whose coefficients at row u and column v is $h_{u,v} = (-1)^{u \cdot v}$ where $u, v \in \mathbb{F}_2^n$. The 2^n–dimensional vectors f and $\widehat{f}$ are have their coordinate x equal to the corresponding function evaluated at x.

Note 2. Matrices indexes may be without ambiguity either an integer or a binary vector being the binary expansion of this integer.

This transform is invertible and the inverse is given by:

$$\widehat{\widehat{f}} = 2^n f \tag{9}$$

When dealing with Boolean functions, we rather resort to the Walsh transform which gets nicer properties than the Fourier transform in most cases. The Walsh transform of a Boolean function f is the Fourier transform of its sign function f_χ where $f_\chi(x) = (-1)^{f(x)} = 1 - 2f(x)$ for $x \in \mathbb{F}_2^n$ that is,

$$\widehat{f_\chi}(u) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) + x \cdot u} \tag{10}$$

As shown in [5], the correspondence between the Fourier and Walsh transforms is given by

$$\forall u \in \mathbb{F}_2^n, \quad \widehat{f_\chi}(u) = 2^n \delta_0(u) - 2\widehat{f}(u), \tag{11}$$

where $\delta_0(u)$ equals 1 if u is the n–dimensional zero vector and equals 0 elsewhere.

The Walsh matrix of any (n, m)–function is the $2^m \times 2^n$ dimensional matrix $W_f = \left(w^f_{u,v}\right)$ with $u \in \mathbb{F}_2^m$ and $v \in \mathbb{F}_2^n$ such that (see [6]):

$$w^f_{u,v} = \sum_{x \in \mathbb{F}_2^n} (-1)^{u \cdot f(x) + v \cdot x} \tag{12}$$

The row u of the matrix W_f is the Walsh transform of the linear combinations of the coordinates of f defined by $x \longmapsto u \cdot f(x)$. The coefficients of the Walsh

matrix of a function is called the spectrum of that function.

Correlation matrices have been defined in [7], they are related to Walsh matrices by a mere normalization coefficient. If R_f is the correlation matrix of f, then

$$R_f = 2^{-n} W_f \tag{13}$$

with coefficients $r^f_{u,v}$. In the paper we usually use correlation matrices to derive the results but it could be done with Walsh matrices as well. The example is given in terms of Walsh matrices.

An interesting property relates the correlation matrices of composed functions.

Proposition 1 (see [6]). *If f is a (n,m)–function and g is a (p,n)–function then*

$$R_{f\circ g} = R_f R_g \tag{14}$$

After these necessary recalls, we are now in position of characterizing the self-synchronization property from a spectral point of view.

4 Spectral Characterization of the Self-Synchronization Property

In this section, we focus on characterizing the self-synchronizing property of the system (5)–(6). As motivated earlier, we can exclusively focus on the next-state function f. It is a $(n+1,n)$–function depending both on the input stream and on the internal state. Let us denote by f^0 (respectively f^1) the (n,n)–function which is the restriction of f to the input bit $c_t = 0$ (respectively to $c_t = 1$). The function f can be expressed as

$$f(c_t, x_t) = \begin{cases} f^0(x_t) & \text{if } c_t = 0 \\ f^1(x_t) & \text{if } c_t = 1 \end{cases} \tag{15}$$

For our purpose, we must define the t^{th} order iterated function of f. It is the $(n+t+1,n)$–function denoted by ϕ_t and defined by

$$\phi_t(c, x_0) = f^{c_t} \circ \cdots \circ f^{c_0}(x_0) \text{ for } t > 0,\ c \in \mathbb{F}_2^{t+1},\ x_0 \in \mathbb{F}_2^n \tag{16}$$

We set $\phi_0 = f$. For a prescribed ciphertext sequence (c) of length $t+1$ and an initial state x_0, the value $\phi_t(c, x_0)$ is the internal state at time $t+1$.
In this section, we characterize the self-synchronizing property in terms of correlation (or Walsh) coefficients. We first focus on self-synchronizing sequences and then apply their properties to address the finite-time and statistical self-synchronization issues of the system (5)–(6).

4.1 Self-synchronizing sequences

Let us denote by $\phi_t^c(x)$ the (n,n)–function which is the restriction of $\phi_t(c,x)$ to a fixed sequence (c) of length $t+1$.

Proposition 2. *The sequence (c) is self-synchronizing if and only if the correlation matrix of ϕ_t^c is a $2^n \times 2^n$ correlation matrix of the form*

$$R_{\phi_t^c} = \begin{pmatrix} 1\,0\cdots 0 \\ \pm 1\,0\cdots 0 \\ \vdots\ \vdots\ \ \vdots \\ \pm 1\,0\cdots 0 \end{pmatrix} \tag{17}$$

Proof. By definition, if (c) is a self-synchronizing sequence, $\phi_t^c(x)$ does not depend on x thus, ϕ_t^c is a constant function. And yet, any linear combination of the coordinate functions of ϕ_t^c is also a constant function. It turns out that any row of (17) is the correlation transform of a constant function. The converse can be derived by using the inverse Fourier transform formula (9).

The matrix $R_{\phi_t^c}$ can easily be determined from the knowledge of the correlation matrices of f^0 and f^1.

Proposition 3. *The expression of the correlation matrix $R_{\phi_t^c}$ is*

$$R_{\phi_t^c} = R_{f^{c_t}} \times \cdots \times R_{f^{c_0}} \tag{18}$$

Proof. The proof is a direct consequence of Proposition 1.

In the following, we use these results to derive some characteristics of the correlation matrices of the functions that have the self-synchronization property.

4.2 Finite-time Self-Synchronization

Let us first notice some important features of correlation matrices. In the sequel, we consider W as a square correlation matrix of dimension $q \times q$.

$$R = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ r_{1,0} & r_{1,1} & \cdots & r_{1,q-1} \\ \vdots & \vdots & & \vdots \\ r_{q-1,0} & r_{q-1,1} & \cdots & r_{q-1,q-1} \end{pmatrix} \tag{19}$$

The matrix W can be rewritten $W = A + N$ with

$$A = \begin{pmatrix} 1 & 0\cdots 0 \\ r_{1,0} & 0\cdots 0 \\ \vdots & \vdots\ \ \vdots \\ r_{q-1,0} & 0\cdots 0 \end{pmatrix} \qquad N = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & r_{1,1} & \cdots & r_{1,q-1} \\ \vdots & \vdots & & \vdots \\ 0 & r_{q-1,1} & \cdots & r_{q-1,q-1} \end{pmatrix}$$

A matrix is said to be of type A if the only non-zero coefficients are located on the first column. A matrix is said to be of type N if all the coefficients of the first row and first column are zero. It is straightforward to verify the following remark:

Remark 3.
- the product of any two matrices of type A is a matrix of type A;
- the product of any matrix of type A with any matrix of type N is a zero matrix;
- the product of any matrix of type N with any matrix of type A is a matrix of the type A;
- the product of any two matrices of type N is a matrix of type N.

Proposition 4. *Consider a Boolean sequence (c) of length $t+1$ and the correlation matrices of the two (n,n)–functions f^0 and f^1: $R_{f^0} = A_{f^0} + N_{f^0}$ and $R_{f^1} = A_{f^1} + N_{f^1}$. The product $R = R_{f^{c_t}} \times \cdots \times R_{f^{c_0}}$ is of type A if and only if the matrix $N_{f^{c_t}} \times \cdots \times N_{f^{c_0}}$ is null.*

Proof.

$$\begin{aligned} R &= R_{f^{c_t}} \times \cdots \times R_{f^{c_0}} \\ &= (A_{f^{c_t}} + N_{f^{c_t}}) \times \cdots \times (A_{f^{c_0}} + N_{f^{c_0}}) \end{aligned} \tag{20}$$

By expanding the expression and using Remark 3, R can be rewritten $R = A+N$ with a type-A matrix A and a type-N matrix $N = N_{f^{c_t}} \times \cdots \times N_{f^{c_0}}$. Because of its structure, A cannot cancel the non-zero coefficients of N. Therefore, R is a type-A matrix if and only if N is null.

The self-synchronization property in the spectral domain can have an algebraic interpretation. It is based on the concept of semigroup.

A semigroup is a set together with an associative internal law. For instance the set of the $2^n \times 2^n$ correlation matrices together with the matrix multiplication is a semigroup. A nilpotent element e is an element such that there exists a large enough positive integer k for which e^k is equal to an absorbing element denoted by 0. A semigroup is said to be generated by a family of elements $E = \{e_0, \ldots, e_n\}$ if any element of the semigroup can be expressed in terms of a product of finite length of elements of E. A nilpotent semigroup is a semigroup that has an absorbing element 0 and in which each element is nilpotent. The nilpotency class of a semigroup S is the smallest positive integer k such that $\forall e \in S$, $e^k = 0$.

Proposition 5. *The system (5)–(6) is finite-time self-synchronizing if and only if the matrices N_{f^0} and N_{f^1} span a nilpotent semigroup.*

Proof. According to Remark 2, a system is finite-time self-synchronizing if and only if there is a positive integer t_c such that any sequence of length greater than t_c is self-synchronizing. That is, in view of Proposition 2, for $t > t_c$, any correlation matrix $R_{\phi_t^c}$ is of type A. The expression of $R_{\phi_t^c}$ given by (18) is,

up to a constant factor, the product of $t+1$ elements of the pair $\{R_{f^0}, R_{f^1}\}$. According to Proposition 4 this product is of type A if and only if whatever is $c \in \mathbb{F}_2^{t+1}$, the product $N_{f^{c_t}} \times \cdots \times N_{f^{c_0}}$ is null. This is the case if and only if the pair $\{N_{f^0}, N_{f^1}\}$ spans a nilpotent semigroup of nilpotency class at most $t+1$.

Now, we aim at pinpointing different classes of self-synchronizing functions. To this end, let us recall an interesting theorem stated in [8] (Theorem 2.1.7).

Theorem 1 (Levitski's theorem). *Any semigroup of nilpotent matrices is triangularizable.*

For any square correlation matrix R of dimension 2^n, let us define its reduced matrix R^* of dimension $(2^n-1) \times (2^n-1)$ which is the matrix R in which the first row and column have been removed.

$$R^* = \begin{pmatrix} r_{1,1} & \cdots & r_{1,q-1} \\ \vdots & & \vdots \\ r_{q-1,1} & \cdots & r_{q-1,q-1} \end{pmatrix}$$

The reduced Walsh matrix W^* of a Walsh matrix W is defined in the same way.

Remark 4. Note that the reduced matrix of N is R^* as well.

Next proposition makes a classification of the possible situations that allow the system (5)–(6) to be finite-time self-synchronizing. It clearly gives a characterization of the functions that can be used in the design of finite-time SSSC.

Proposition 6. *The system* (5)–(6) *with the next-state function f (and the associated (n,n)–functions f^0 and f^1) is finite-time self-synchronizing if and only if the reduced correlation matrices $R^*_{f^0}$ and $R^*_{f^1}$ are nilpotent and fulfill one of the following cases:*

Case 1 Both matrices $R^*_{f^0}$ and $R^*_{f^1}$ are lower triangular.

Case 2 Both matrices $R^*_{f^0}$ and $R^*_{f^1}$ are not lower triangular but can be simultaneously triangularized by a change of basis whose matrix is the reduced correlation matrix R^*_p of some (n,n)–function p. This matrix has to be invertible. In this situation, the following equalities hold: $R^*_p R^*_{f^0}(R^*_p)^{-1} = \widetilde{R}^*_{f^0}$ and $R^*_p R^*_{f^1}(R^*_p)^{-1} = \widetilde{R}^*_{f^1}$ with $\widetilde{R}^*_{f^0}$ and $\widetilde{R}^*_{f^1}$ two lower triangular matrices with zeros on the diagonal.

Case 3 Both matrices $R^*_{f^0}$ and $R^*_{f^1}$ are not lower triangular. They can be however simultaneously triangularized like in Case 2. but unlike Case 2, R^*_p does not correspond to a correlation matrix.

Proof. Proposition 5 states that the system (5)–(6) is finite-time self-synchronizing if and only if N^0_f and N^1_f span a nilpotent semigroup. In view of Remark 4, the same holds for the matrices $R^*_{f^0}$ and $R^*_{f^1}$. Then, in view of Theorem 1, they can be simultaneously triangularized. Cases 1, 2 and 3 are exclusive and describe all the possible situations.

The following lemma is required in order to interpret the significance of Proposition 6.

Lemma 1. *For any bijection p of $\mathbb{F}_2^n$ the relation $(R_p^*)^{-1} = R_{p^{-1}}^*$ holds.*

Proof. It can be seen from (14) that $R_{p^{-1}} = (R_p)^{-1}$. Since p is an invertible transformation we know from [7] that its correlation matrix is orthogonal. If we denote the transpose of R by R^t then,

$$(R_p^*)^{-1} = (R_p^*)^t = (R_p^t)^* = (R_p^{-1})^* = (R_{p^{-1}})^*$$

Case 1 corresponds to the case when f^0 and f^1 are strict T–functions. Indeed, the reduced correlation matrix is lower triangular with zeros on diagonal except on the first row if and only if the corresponding function is a strict T–function (see Proposition 11 in [9]). Therefore Case 1 refers to functions which have been already proposed through the open literature.
Case 2 corresponds to the situation when f^0 and f^1 are not strict T–functions but functions of the form $f^0 = p \circ \widetilde{f}^0 \circ p^{-1}$ and $f^1 = p \circ \widetilde{f}^1 \circ p^{-1}$ where $\widetilde{f}^0$ and $\widetilde{f}^1$ are strict T–functions and p a bijection over $\mathbb{F}_2^n$. A consequence of Lemma 1 is that this case is nothing but Case 1 in which the functions f^0 and f^1 have been both right-composed with the same bijective function p and left composed with p^{-1}. Thus, this case is equivalent to Case 1 up to an invertible transformation of the internal state.
Case 3 corresponds to self-synchronizing functions that are not based on strict T–functions. This case is the most interesting one insofar as it allows to identify new classes of self-synchronizing functions. An example of such a function is given in Section 6.

Remark 5. It is interesting to note that the synchronization delay t_c precisely corresponds to the nilpotency class of the semigroup spanned by $R_{f^0}^*$ and $R_{f^1}^*$. Moreover, since Cases 1 and 2 are based on strict T–functions, the maximum nilpotency class is bounded by n in these situations. In Case 3 the maximum nilpotency class is the dimension of the matrices which is $2^n - 1$. Therefore, if two reduced correlation matrices $R_{f^0}^*$, $R_{f^1}^*$ span a nilpotent semigroup of nilpotency class greater than n, it necessary corresponds to Case 3.

The problem of determining if any two (n, n)–functions f^0 and f^1 can be used to design finite-time self-synchronizing systems as defined by (5)–(6) amounts to checking whether or not their reduced correlation matrices $R_{f^0}^*$ and $R_{f^1}^*$ span a nilpotent semigroup. From Proposition 5, if this is the case they can be simultaneously triangularized. The book [8] provides interesting approaches to determine whether or not a set of matrices can be simultaneous triangularized. An algorithm that simultaneously triangularizes a set of matrices is given in the paper [10]. The algorithm can be applied to any set of matrices, it simply fails when no common triangularization basis exists.

Extension to multi-bit symbols We now explains how to extend these results to streams whose symbols are composed of more than one bit, say k bits for instance. In this situation, m_t, c_t, $\widehat{m}_t$ and z_t belong to $\mathbb{F}_2^k$. The next state function is now $f : \mathbb{F}_2^k \times \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$ and the output function is $h : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^k$. In order to extend the previous results it is necessary to do the same decomposition of f than in (15). This time, f can be viewed as a set of 2^k functions $f^i : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^n$ for $i \in \mathbb{F}_2^k$. All the previous results can be extended. It suffices to consider the 2^k matrices R_{f^i} instead of only two matrices. Essentially, the following result holds

Proposition 7. *The system is finite-time self-synchronizing if and only if the set of reduced correlation matrices $R^*_{f^i}$ span a nilpotent semigroup.*

Proof. The proof is the same as in Proposition 5 except that there are now 2^k matrices to consider instead of only 2.

The classification of proposed in Proposition 6 still holds.

4.3 Statistical Self-Synchronization

In this paragraph, in order to enlarge the class of potential candidate functions, we relax the finite-time self-synchronization constraint and extend Definition 4. Indeed, in practice, it is acceptable that the synchronization delay t_c is not bounded, but may be a random variable with a probability law that decreases to zero as time goes to infinity. In other words, the probability of being synchronized reaches one while the length of the stream (c) increases. This concept is viable only if the probability of being synchronized is sufficiently close to one for some reasonable length. Such systems are called statistical SSSC. If (c) is a random sequence then, the synchronization delay t_c is a random variable. In such a case, it is denoted by T_c.

Definition 6 (Statistical self-synchronization). *The system* (5)–(6) *is statistically self-synchronizing if* $\lim_{t \to +\infty} \Pr(T_c \leq t) = 1$. *The random variable T_c is called the random synchronization delay for the random sequence* (c).

Remark 6. It is interesting to note that if the probability of synchronization is one for some constant delay, Definition 6 reduces to Definition 4. Therefore, finite-time self-synchronization is nothing but a special case of statistical self-synchronization.

5 State Probability

Ensuring the self-synchronizing property is a first feature required for the design of SSSC. The security has to be further assessed. From this perspective, we think it is interesting to determine the probability that a given state can be reached

after a fixed number of iterations. A cryptographic criterion is that at the decryptor side, all the states are reached with almost the same probability, ideally the same, for a random cryptogram stream symbols. Studying this criterion is the purpose of this section.

Let us first consider the following matter. We are given a (n, m)–function g and a random variable $X \in \mathbb{F}_2^n$ whose value is described by the probability law $p : \mathbb{F}_2^n \longrightarrow \mathbb{R}$ defined by $p(x) = \Pr[X = x]$. We want to know the probability law $q : \mathbb{F}_2^m \longrightarrow \mathbb{R}$ that describes the random variable $Y \in \mathbb{F}_2^m$ defined by $Y = g(X)$. The function q is defined by $q(y) = \Pr[y = g(X)]$. Without ambiguity, the notation p (respectively q) refers either to the function or to the 2^n (respectively 2^m) column vector whose coordinate index $x \in \mathbb{F}_2^n$ (respectively $y \in \mathbb{F}_2^m$) has the value $p(x)$ (respectively $q(y)$). The same holds for $\widehat{p}$ and $\widehat{q}$ which are the Fourier transforms of p and q.

Proposition 8. *Let R_g be the correlation matrix of g. Applying the function g to a variable whose value is chosen according to the probability law described by p gives a vector whose value is described by the probability law q. They are related by the relation*

$$\widehat{q} = R_g \widehat{p} \tag{21}$$

Proof. Let us first relate q and p.

$$\begin{aligned} q(y) &= \textstyle\sum_{x \in \mathbb{F}_2^n | g(x) = y} p(x) \\ &= 2^{-m} \textstyle\sum_{x \in \mathbb{F}_2^n} p(x) \sum_{u \in \mathbb{F}_2^m} (-1)^{u(g(x)+y)} \\ &= 2^{-m} \textstyle\sum_{u \in \mathbb{F}_2^m} \sum_{x \in \mathbb{F}_2^n} (-1)^{u \cdot y} p(x) (-1)^{u \cdot g(x)} \end{aligned}$$

We now express the Fourier transform of q.

$$\begin{aligned} \widehat{q}(s) &= \textstyle\sum_{y \in \mathbb{F}_2^m} q(y) (-1)^{s \cdot y} \\ &= 2^{-m} \textstyle\sum_{u \in \mathbb{F}_2^m, x \in \mathbb{F}_2^n} \underbrace{\sum_{y \in \mathbb{F}_2^m} (-1)^{u \cdot y + s \cdot y}}_{\begin{cases} 2^m & \text{if } u = s \\ 0 & \text{else} \end{cases}} p(x) (-1)^{u \cdot g(x)} \\ &= \textstyle\sum_{x \in \mathbb{F}_2^n} p(x) (-1)^{s \cdot g(x)} \\ &= 2^{-n} \textstyle\sum_{x \in \mathbb{F}_2^n} p(x) \sum_{z \in \mathbb{F}_2^n} (-1)^{s \cdot g(z)} \sum_{v \in \mathbb{F}_2^n} (-1)^{v \cdot (x+z)} \\ &= \textstyle\sum_{v \in \mathbb{F}_2^n} \underbrace{\sum_{x \in \mathbb{F}_2^n} p(x) (-1)^{v \cdot x}}_{\widehat{p}(v)} \underbrace{2^{-n} \sum_{z \in \mathbb{F}_2^n} (-1)^{s \cdot g(z) + v \cdot z}}_{r^g_{s,v} = 2^{-n} w^g_{s,v}} \end{aligned}$$

where $w^g_{s,v}$ is as defined by (12). The result holds.

The following corollary can be stated

Corollary 1.

$$q = H^{-1} R_g H p \tag{22}$$

Proof. Equation (21) also reads $Hq = R_g Hp$. The result holds.

If we restrict the vector p of Proposition 8 to the uniform probability vector the following corollary, which corresponds to Lemma 1 in [11], is direct.

Corollary 2. *Let X follow the uniform distribution and g be a (n,n)–function. The probability distribution, after applying the function g to X reads*

$$\forall x \in \mathbb{F}_2^n,\ \Pr[g(X) = x] = 2^{-n} \sum_{s \in \mathbb{F}_2^n} (-1)^{s \cdot x} r^g_{s,0} \tag{23}$$

where $r^g_{s,0}$ is the coefficient of the correlation matrix of g of the s^{th} row and of the first column.

Till now, we have always considered that the initial state x_0 is chosen according to the uniform distribution. We have also assumed that the symbols of the ciphertext stream (c) are uniformly distributed. Let us stress that even though this assumption makes sense in cryptography since the stream (c) should not be distinguishable from a true uniform random stream, it should be considered with caution. Indeed, the uniformity of (c) depends on the uniformity of (z) which in turn depends on the function f and h.
We now focus on the evolution of the probability law modified by the next-state function f.

Proposition 9. *Let (C) be a uniform random sequence and assume a uniform random distribution of the initial state X_0. Then, the probability that the iterated function ϕ_t^C returns the state $x \in \mathbb{F}_2^n$ is*

$$P[\phi_t^C(X_0) = x] = \frac{1}{2^{n+t+1}} \sum_{s \in \mathbb{F}_2^n} (-1)^{s \cdot x} \left[\left[R_{f^0} + R_{f^1} \right]^{t+1} \right]_{s,0} \tag{24}$$

Proof. Since (C) is a uniform random sequence of length $t+1$, the probability of having this specific sequence in $t+1$ iterations is 2^{-t-1}.

$$P[\phi_t^C(X_0) = x] = \frac{1}{2^{t+1}} \sum_{c \in \mathbb{F}_2^{t+1}} P[f^{C_t} \circ \cdots \circ f^{C_0} = x]$$

Then, in view of Proposition 1 and Corollary 2

$$\begin{aligned} P[\phi_t^C(X_0) = x] &= \frac{1}{2^{t+1}} \sum_{c \in \mathbb{F}_2^{t+1}} \frac{1}{2^n} \sum_{s \in \mathbb{F}_2^n} (-1)^{s \cdot x} \left[R_{f^{C_t}} \times \cdots \times R_{f^{C_0}} \right]_{s,0} \\ &= \frac{1}{2^{n+t+1}} \sum_{s \in \mathbb{F}_2^n} (-1)^{s \cdot x} \left[\left[R_{f^0} + R_{f^1} \right]^{t+1} \right]_{s,0} \end{aligned}$$

Proposition 10. *Assuming a random sequence (C) of length $t+1$ and a random initial state X_0, the system (5)–(6) has an equal probability to be in each state if and only if the first column of the matrix $\left[R_{f^0} + R_{f^1}\right]^{t+1}$ denoted by r_0 is given by*

$$r_0 = \left(2^{t+1}\ 0 \cdots\ 0\right)^T \tag{25}$$

Proof. Proving this result amounts to solving a linear algebra problem. Let ν be the 2^n–dimensional column vector whose coefficients at row x is the probability of being in the state x. In our case, we set the value of each coefficient to 2^{-n}. Considering Proposition 9, denoting by H the 2^n–dimensional Hadamard matrix defined by $H = (h_{s,x}) = (-1)^{s \cdot x}$ for $s, x \in \mathbb{F}_2^n$ and by k the constant 2^{-n-t-1}, the problem reads

$$\nu = kHr_0$$

where r_0 is the unknown. Since both k and H are invertible, the system can be solved and has a unique solution.

Remark 7. The fact that each state is reached with an equal probability for a random sequence of length $t+1$ does not mean that each state is reached with an equal probability with a uniform random sequence of length $t+2$.

Remark 8. Proposition 10 states that assuming a uniform distribution of the initial state X_0 and a uniform random sequence (C), the uniform distribution of the internal state at time t is achieved if and only if the first column vector of $\left[R_{f^0} + R_{f^1}\right]^{t+1}$ is given by the relation (25). Under this condition, a uniform distribution is achieved at any time if and only if f is balanced.

Since the first column of $R_{f^0} + R_{f^1}$ is the same as the first column of R_f, if f is balanced, the condition $r_{s,0}^{f^0} = -r_{s,0}^{f^1}$ if $s \neq 0$ holds.

6 Example

As an illustration of the finite-time SSSC described by Case 3 in Section 4.2, let us show that the set of functions described by Case 3 is not empty. In this example we rather use Walsh matrices than correlation matrices, it allows to have matrices with integer coefficients. We recall that the correspondence between these two kind of matrices is given by (13). We consider $n = 3$. The next-state function f is based on two almost-bent functions f^0 and f^1 as defined by (15):

$$\begin{cases} f_0^0(x) = 1 + x_0 + x_1 + x_0x_2 \\ f_1^0(x) = 1 + x_0x_1 + x_2 + x_0x_2 \\ f_2^0(x) = x_1x_2 \end{cases} \qquad \begin{cases} f_0^1(x) = x_0x_1 + x_2 \\ f_1^1(x) = 1 + x_0 + x_0x_1 + x_1x_2 \\ f_2^1(x) = 1 + x_0 + x_0x_1 + x_0x_2 \end{cases}$$

Then, the reduced Walsh matrices can be worked out by using (12)

$$W^*_{f^0} = \begin{pmatrix} 0 & -4 & -4 & 0 & 0 & 4 & -4 \\ 0 & -4 & 4 & -4 & -4 & 0 & 0 \\ 0 & 0 & 0 & -4 & 4 & 4 & 4 \\ 0 & 4 & 0 & 4 & 0 & -4 & 0 \\ 0 & 0 & -4 & 4 & 0 & 0 & -4 \\ 0 & 0 & 4 & 0 & -4 & -4 & 0 \\ 0 & 4 & 0 & 0 & 4 & 0 & 4 \end{pmatrix} \quad W^*_{f^1} = \begin{pmatrix} 0 & 0 & 0 & 4 & 4 & 4 & -4 \\ -4 & 0 & -4 & -4 & 0 & 4 & 0 \\ -4 & 0 & 4 & 0 & -4 & 0 & -4 \\ -4 & 0 & 0 & 0 & 0 & 4 & -4 \\ -4 & 0 & 0 & -4 & -4 & 0 & 0 \\ 0 & 0 & 4 & 4 & 0 & 0 & -4 \\ 0 & 0 & -4 & 0 & 4 & 4 & 0 \end{pmatrix}$$

According to Theorem 1, the matrices $W^*_{f^0}$ and $W^*_{f^1}$ span a nilpotent semigroup. Indeed, they can be simultaneously triangularized and the algorithm of the paper [10] allows to find out one possible change of basis. The matrix

$$W^*_p = \begin{pmatrix} 1 & -1 & -2 & -2 & 4 & 0 & 0 \\ 1 & -1 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4 & -2 & 2 & 0 \\ 0 & 2 & 1 & -3 & 1 & -1 & -1 \\ -1 & -1 & 1 & -3 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -3 & -1 & 1 \\ 0 & 2 & 1 & 1 & 3 & 1 & 1 \end{pmatrix}$$

triangularizes both $W^*_{f^0}$ and $W^*_{f^1}$.
It can be checked that the class of nilpotency of this semigroup is larger than n and because of Remark 5 there are no bijection from $\mathbb{F}_2^3$ that allows to triangularize the semigroup. Therefore this system corresponds to Case 3.

The algebraic normal form of f is

$$\begin{cases} f_0(c,x) = 1 + x_0 + x_1 + x_0x_2 + c + x_0c + x_1c + x_0x_1c + x_2c + x_0x_2c \\ f_1(c,x) = 1 + x_0x_1 + x_2 + x_0x_2 + x_0c + x_2c + x_0x_2c + x_1x_2c \\ f_2(c,x) = x_1x_2 + c + x_0c + x_0x_1c + x_0x_2c + x_1x_2c \end{cases}$$

This function is balanced, in view of Remark 8 each state is reached with the same probability.

7 Conclusion

Two kinds of self-synchronization have been defined. Finite-time self-synchronization has been characterized from the spectral analysis point of view. It has been shown that it is possible to achieve finite-time self-synchronization using functions which are not strict T–functions. Three cases have been pinpointed. The known strict T–function case, the case when strict T–functions have been right and left composed with a permutation and its inverse and the case which is not based on strict T–functions. The latter case is interesting due to its novelty, an algebraic characterization in terms of nilpotent semigroups has been performed. The example encourages to study the problem from this point of view since it

shows that Case 3 contains functions that are worth to be considered. We then have discussed statistical self-synchronization as a generalization of finite-time self-synchronization.
These characterizations will provide constructive material in order to find out classes of keyed families of functions for cryptographic purposes. The way how to incorporate performance constraints regarding the security deserves a deeper insight in the perspective of designing fully specified self-synchronizing stream ciphers.

References

1. J. Daemen. *Cipher and Hash function design, strategies based on linear and differential cryptanalysis.* PhD Thesis, Katholieke Universiteit Leuven, 1995.
2. A. Joux and F. Muller. Chosen-ciphertext attacks against mosquito. *Fast Software Encryption, Lecture Note in Computer Science*, 4047:87–99, Springer 2006.
3. A. Klimov and A. Shamir. A new class of invertible mappings. In B. Kaliski, Ç. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 470–483. Springer Berlin / Heidelberg, 2003. 10.1007/3-540-36400-5_34.
4. A. Klimov and A. Shamir. Cryptographic applications of t-functions. In M. Matsui and R Zuccherato, editors, *Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 248–261. Springer Berlin / Heidelberg, 2004. 10.1007/978-3-540-24654-1_18.
5. C. Carlet. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, chapter Boolean Functions for Cryptography and Error-Correcting Codes. In [12], 2010.
6. C. Carlet. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, chapter Vectorial Boolean Functions for Cryptography. In [12], 2010.
7. J. Daemen, R. Govaerts, and V. Joos. Correlation matrices. In *Fast Software Encryption : Second International Workshop, LNCS 1008*, pages 275–285. Springer-Verlag, 1994.
8. H. Radjavi and P. Rosenthal. *Simultaneous Triangularization.* Springer, 2000.
9. J. Parriaux, P. Guillot, and G. Millérioux. Synchronization of boolean dynamical systems: A spectral characterization. In C. Carlet and A. Pott, editors, *Sequences and Their Applications SETA 2010*, volume 6338 of *Lecture Notes in Computer Science*, pages 373–386. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-15874-2_32.
10. C. Dubi. An algorithmic approach to simultaneous triangularization. *Linear Algebra and its Applications*, 430(11-12):2975 – 2981, 2009.
11. K. Nyberg and M. Hermelin. Multidimensional walsh transform and a characterization of bent functions. *Proceedings of the 2007 IEEE Information Theory Workshop on Information Theory for Wireless Networks*, 2007.
12. Y. Crama. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering.* Cambridge Press, 2010.