Designing self-synchronizing switched linear systems: an application to communications

Jeremy Parriaux, Gilles Millérioux

Designing self-synchronizing switched linear systems: an application to communications

Jérémy Parriaux[a], Gilles Millérioux[a]

[a] *Nancy University*
Research Center for Automatic Control of Nancy (CRAN UMR 7039)
2 rue Jean Lamour, 54519 Vandoeuvre-les-Nancy, France,
(e-mail: jeremy.parriaux@esstin.uhp-nancy.fr, gilles.millerioux@esstin.uhp-nancy.fr)

Abstract

This paper addresses the problem of self-synchronizing dynamical systems in a so-called master-slave configuration. The study is motivated by potential cryptographic applications. It is shown that the notion of flatness is central for guaranteeing a finite-time self-synchronization and that the concept of transmission zero plays also an important role. Next, the finite-time synchronization is relaxed to give rise to a so-called statistical self-synchronization, a mode of operation which makes sense in classical cryptography which operates over finite fields. The fact that switched linear systems are of great interest in this context is motivated.

Keywords: switched systems; synchronization; communication; flatness; invertibility

1. Introduction

Synchronization of dynamical systems is an important purpose in many fields like biology, mechanics, communications. Synchronization means coordinated behavior of different interconnected entities involved in an overall system. Many different definitions and related configurations, in terms of coupling, can be investigated. An exhaustive and interesting overview can be found in [2]. A special kind of synchronization is the self-synchronization. By self-synchronization, it is meant a coordinated behaviour which is achieved without any external control.

The configuration under consideration in this paper is a master-slave configuration with unidirectional coupling. It is borrowed from the field of communications and more specifically secure transmissions. In this context, cryptography plays a central role. It is the discipline which is mainly intended to protect information and to guarantee confidential exchanges through public channels. Since the 90's, many "scrambling" methods resorting to synchronized chaotic dynamical systems have been proposed. In the works [4, 10, 9, 1], it is highlighted the connection between cryptography and the use of synchronized dynamical systems in a master-slave configuration exhibiting complex dynamics. The exhogenous input of the master dynamical system is the information to

be encrypted. The master plays the role of the cipher. The slave is a dynamical system which plays the role of the decipher. The coupling is achieved through the output of the cipher which acts as the cryptogram. In all the "scrambling" methods, synchronization between the master and the slave is guaranteed without any external control. In other words, self-synchronization is achieved by means of the output coupling. That corresponds to some classical communication setups for which insertion of synchronization flags in the transmitted packets is forbidden for throughput purposes.

As it turns out, most of the "scrambling" methods resort to observer-based approaches for ensuring the synchronization and in general, asymptotical synchronization is achieved. It is clear that the asymptotical convergence may appear when operating over the field of real numbers as it is the case for chaotic systems. On the other hand, it makes no longer sense when operating over finite fields as it is precisely the case in classical cryptography. In [10], it has been shown that a finite-time synchronization can be achieved whenever the dynamical system playing the role of the cipher is flat. In this case, the communication scheme is structurally equivalent to a so-called classical *self-synchronizing stream cipher*. As a result, resorting to flat dynamical systems not only makes sense from a cryptographic point of view but would provide a new approach for the design of *self-synchronizing stream cipher*s. However, this study was reduced to analysis.

The aim of the present work is to provide a constructive approach for the design of dynamical systems having the self-synchronization property. Discrete-time switched linear systems are specifically addressed because they correspond to the so-called Maiorana McFarland construction which has proved to produce functions that have many interesting cryptographic properties (see [3]). Then, the finite-time convergence will be relaxed to give rise to the so-called statistical self-synchronizing stream ciphers. Such ciphers still make sense in classical cryptography but has only been touched on so far. Hence, the constructions proposed in this paper can be considered as a first step towards a complete framework for designing new classes of self-synchronizing stream ciphers. Let us point out that we mainly focus on the structural considerations of the ciphers while disregarding the security aspects which would be here out of the scope and are discussed in companion papers.

The outline of this paper is the following. In Section 2, strict necessary background on cryptography is provided. The role of self-synchronization in this context is emphasized and a formal definition of finite-time self-synchronization is given. In Section 3, the design of admissible master-slave configurations, described by piecewise linear systems, achieving finite-time self-synchronization is detailed. A constructive approach for guaranteeing the self-synchronization is suggested. It is mainly based on the notion of nilpotent semigroups. A connection between the issue of guaranteeing self-synchronization and the concept of flatness is brought out. Further considerations for the design are developed in

Section 4 where it is shown that the concept of transmission zero of a dynamical system plays an important role as well. Section 5 extends finite-time self-synchronization to statistical self-synchronization by releasing some constraints. Finally, Section 6 is devoted to illustrative examples.

Notation: $\mathbf{1}_n$ stands for the identity matrix of dimension n, $\mathbf{0}$ stands for the zero matrix of appropriate dimension regarding the situation. We denote by $\{z\}_{k_1}^{k_2}$ the sequence $\{z_{k_1}, \ldots, z_{k_2}\}$ when the initial and final times k_1 and k_2 are defined, otherwise the sequence is merely denoted by $\{z\}$.

2. Cryptography and synchronization

2.1. Background on cryptography

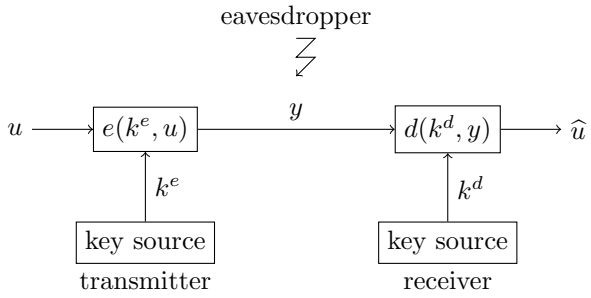

Figure 1: General encryption mechanism

A general encryption mechanism, also called cryptosystem or cipher, is depicted in Figure 1. We are given an alphabet A that is, a finite set of elements named symbols. On the *transmitter* part, a plaintext (also called information or message) $\{u\} \in \mathcal{U}$ ($\mathcal{U}$ is called the message space) consisting of a string of symbols $u_k \in A$ is encrypted according to an encryption function e which depends on the key $k^e \in \mathcal{K}$ ($\mathcal{K}$ is called the key space). The resulting ciphertext $\{y\} \in \mathcal{C}$ ($\mathcal{C}$ is called the ciphertext space), a string of symbols $y_k \in B$, B being a set usually identical to A, is conveyed through a public channel to the *receiver*. At the receiver side, the ciphertext y_k is decrypted according to a decryption function d which depends on the key $k^d \in \mathcal{K}$. For a prescribed k^e, the function e must be invertible. Cryptography distinguishes asymmetric and symmetric ciphers. Asymmetric cryptography is largely based upon computationally very demanding mathematical problems, for instance, integer factorization into primes. It is not discussed in this paper.

In symmetric encryption, both keys are identical that is, $k^d = k^e$. That explains the terminology "symmetric". This kind of encryption obeys a master-slave configuration. The transmitter, that is the master, is called in this context the cipher. It delivers a complex sequence (theoretically indistinguishable from a uniformly random one) used to conceal information. The information to be

kept secret is, in some sense, "mixed" with the complex sequence so that the resulting sequence (called the cryptogram) conveyed to the receiver, cannot be understood by any unauthorized party. For proper information recovery, the receiver, that is the slave, called in this context the decipher, must deliver the same complex sequence synchronized with the cipher.

For stream ciphers depicted in Figure 2, the keys k^e and k^d are replaced by time-varying sequences called *running keys* or *key-streams*. They are denoted by $\{x\}$ (with samples x_k) at the transmitter part and by $\{\widehat{x}\}$ (with samples $\widehat{x}_k$) at the receiver part. As a result, stream ciphers require key-stream generators at both ends. The key-streams $\{x\}$ and $\{\widehat{x}\}$ must be synchronized to guarantee the equality $x_k = \widehat{x}_k$ and thereby to match the symmetry principle. The secret keys k^e and k^d are some suitable selected parameters of the respective key-stream generators, the selection being based on security considerations. As mentioned in the introduction, some applications require that the synchro-

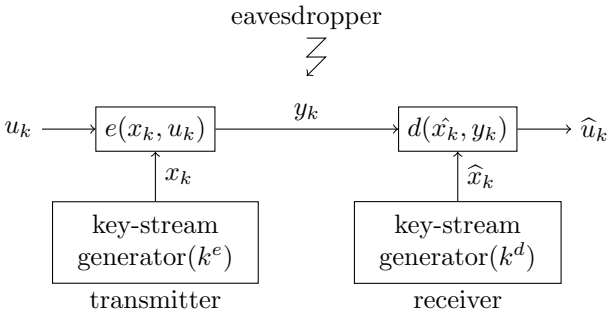

Figure 2: Stream cipher

nization is guaranteed without any external control that is, self-synchronization must be achieved. In such a case, the stream ciphers must have a special architecture and they are called Self-Synchronizing Stream-Ciphers, SSSC for short. An overview about this class of ciphers can be found in [6, 9].

2.2. Self-synchronization and ciphering

Self-Synchronizing Stream Ciphers admit at the transmitter and receiver ends the respective equations:

$$\begin{cases} x_k = g_{k^e}(y_{k-K}, \ldots, y_{k-1}) \\ y_k = e(x_k, u_k) \end{cases} \tag{1}$$

$$\begin{cases} \widehat{x}_k = g_{k^d}(y_{k-K}, \ldots, y_{k-1}) \\ \widehat{u}_k = d(\widehat{x}_k, y_k) \end{cases} \tag{2}$$

where g_{k^e} and g_{k^d} are the functions that generate the key-streams $\{x\}$ and $\{\widehat{x}\}$. Both functions depend on the last K past values of y_k.

The ciphertext y_k is worked out through an encryption function e which must be invertible for any prescribed x_k. The decryption is performed through a function d depending on the ciphertext y_k and on the running key $\widehat{x}_k$ of the receiver. Such a function must obey the rule:

$$\widehat{u}_k = d(\widehat{x}_k, y_k) = u_k \text{ if } \widehat{x}_k = x_k \tag{3}$$

According to (3), the synchronization of the key-streams $\{x\}$ and $\{\widehat{x}\}$ generated respectively at the transmitter and receiver sides is a condition for proper decryption. Since the function g_{k^e} is identical at the transmitter and receiver sides and share the same arguments, namely the past ciphertexts y_{k-i} $(i = 1, \ldots, K)$, it is clear that the generators synchronize automatically after a finite transient time of length K. This kind of self-synchronization is called finite-time self-synchronization. A more formal definition will be given a little bit later.

Actually, the model (1)–(2) of an SSSC is a conceptual model, called canonical representation, that corresponds to different architectures. In particular, it admits an equivalent recursive form involving a K–dimensional internal state $z_k = (y_{k-K}, \ldots, y_{k-1})$. Its i^{th} coordinate is denoted by $(z_k)_i$. The equations of the recursive form read

$$\begin{cases} (z_{k+1})_i & = (z_k)_{i-1} \text{ if } i > 0,\ y_k \text{ if } i = 0 \\ y_k & = e(g_{k^e}(z_k), u_k) \end{cases} \tag{4}$$

$$\begin{cases} (\widehat{z}_{k+1})_i & = (\widehat{z}_k)_{i-1} \text{ if } i > 0,\ y_k \text{ if } i = 0 \\ \widehat{u}_k & = d(g_{k^d}(\widehat{z}_k), y_k) \end{cases} \tag{5}$$

The state updating transformation of the canonical recursive form (4)–(5) is a mere shift fed with the previous ciphertexts.

It turns out that resorting to dynamical systems instead of implementing directly the canonical form (1)–(2) or its equivalent recursive form (4)–(5) would broaden the possibilities of design of stream ciphers. More formally, we should propose a setup with two parts as shown in Figure 3. The first part (playing the role of the cipher) consists of a dynamical system $\mathcal{C}$, with input u_k (playing the role of the plaintext), output y_k (playing the role of the cipher) and state vector x_k (playing the role of the key-stream).

$$\mathcal{C} \begin{cases} x_{k+1} & = f(x_k, u_k) \\ y_k & = h(x_k, u_k) \end{cases} \tag{6}$$

The output y_k ensures a unidirectional coupling with the second part, the dynamical system $\mathcal{D}$ (playing the role of the decipher) with state vector $\widehat{x}_k$. The quantity y_k acts as an input for $\mathcal{D}$.

$$\mathcal{D} \begin{cases} \widehat{x}_{k+1} & = \widehat{f}(\widehat{x}_k, y_k) \\ \widehat{u}_k & = \widehat{h}(\widehat{x}_k, y_k) \end{cases} \tag{7}$$

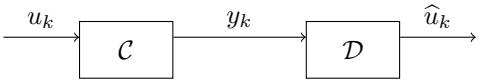

Figure 3: Dynamical system-based cryptosystems

The symbol $\widehat{u}_k$ is the output of $\mathcal{D}$. In a cryptographic context this is the recovered information, it must be equal to u_k whenever $x_k = \widehat{x}_k$. The recursive form (4)–(5) is actually a special case of the set-up (6)–(7). However, not all the dynamical systems can be candidate. They must have the self-synchronization property which obeys the following definition:

Definition 1 (Finite-time self-synchronization)**.** *The unidirectional coupled system $\mathcal{C}$–$\mathcal{D}$ is finite-time self-synchronizing if, for all admissible input sequences,*

$$\exists K \in \mathbb{N},\, \forall x_0,\, \widehat{x}_0,\, \forall k \geq K,\ x_k = \widehat{x}_k \tag{8}$$

More generally, a delay $r \in \mathbb{N}$ can be allowed. If so, (8) turns into

$$\exists K \in \mathbb{N},\, \forall x_0,\, \widehat{x}_0,\, \forall k \geq K,\ x_k = \widehat{x}_{k+r} \tag{9}$$

Finally, the issue to be investigated is the following. How to design a master-slave setup $\mathcal{C}$–$\mathcal{D}$ so that

- self-synchronization (8) (possibly (9)) can be guaranteed?
- proper input recovery $\widehat{u}_k = u_k$ is ensured whenever self-synchronization is achieved?

It is the purpose of the next sections. We concentrate on the special class of switched linear systems. Indeed, when symmetric cryptography is sought, the functions to be considered are often the Boolean ones, that is, for some positive integers n and m the functions $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^m$ where $\mathbb{F}_2$ denotes the two-element field. And yet, it turns out that switched linear systems correspond to the Maiorana McFarland construction which has proved to provide functions that have many interesting cryptographic properties like the highest nonlinearity, high correlation immunity and good propagation characteristics [3]. Nevertheless, in digital transmissions, it can be interesting to consider other finite-fields than $\mathbb{F}_2$ in general, we denote a finite field by $\mathbb{F}$. The cardinality is no longer exclusively 2 but is p^q with p a prime and q a positive integer. When $q = 1$, all the operations, namely, addition, subtraction, multiplication and inversion are still defined like in the field of real numbers except that the results are computed modulo p.

3. Finite-time self-synchronization and switched systems

The equations of the setup read at the transmitter part

$$\mathcal{C} \left\{ \begin{array}{rcl} x_{k+1} & = & A_{\sigma(k)} x_k + B_{\sigma(k)} u_k \\ y_k & = & C_{\sigma(k)} x_k + D_{\sigma(k)} u_k \end{array} \right. \tag{10}$$

and at the receiver part

$$\mathcal{D}\left\{\begin{array}{lcl}\widehat{x}_{k+1} & = & A'_{\sigma(k)}\widehat{x}_k + B'_{\sigma(k)}y_k \\ \widehat{u}_k & = & C'_{\sigma(k)}\widehat{x}_k + D'_{\sigma(k)}y_k\end{array}\right. \tag{11}$$

with $u_k, \widehat{u}_k \in \mathbb{F}$, $y_k \in \mathbb{F}$ and $x_k, \widehat{x}_k \in \mathbb{F}^n$.

The switching function σ is defined as

$$\sigma : k \in \mathbb{N} \mapsto j = \sigma(k) \in \{1, \ldots, J\} = \mathcal{J}$$

At a given time k, the index j corresponds to the mode of the system given by the switching function σ. The number of modes is denoted by J. All the matrices, namely $A_{\sigma(k)} \in \mathbb{F}^{n\times n}$, $B_{\sigma(k)} \in \mathbb{F}^{n\times 1}$, $C_{\sigma(k)} \in \mathbb{F}^{1\times n}$ and $D_{\sigma(k)} \in \mathbb{F}$ belong to the respective finite sets $\{A_j,\, j \in \mathcal{J}\}$, $\{B_j,\, j \in \mathcal{J}\}$, $\{C_j,\, j \in \mathcal{J}\}$ and $\{D_j,\, j \in \mathcal{J}\}$. The switching function must depend on the output y_k. The motivation of such a dependence lies in that the switching rule must also be self-synchronizing. Thus, it must depend on shared variables and so on the output y_k or a finite sequence of delayed outputs. It is worth pointing out that the writing $\sigma(k)$ is thereby somehow abusive. We denote by $\{v\}$ the sequence of modes $\{v\} = \{\sigma(k), \sigma(k+1), \ldots\}$ and the i^{th} element is denoted by v_i. If the sequence $\{v\}$ has a finite length K, it is an element of the set denoted by $\mathcal{J}^K$. In the following, we derive conditions for guaranteeing self-synchronization of the master-slave setup (10)–(11) and propose constructive approaches for achieving finite-time self-synchronization.

3.1. General conditions

Theorem 1. *The setup* (10)–(11) *is finite-time self-synchronizing whenever the three following conditions are fulfilled:*

- $\forall j \in \mathcal{J},\ D'_j \neq 0$ (12)

- $\exists K \in \mathbb{N}, \forall x_0,\, \widehat{x}_0,\, \forall \{v\} \in \mathcal{J}^K,\ \prod_{i=0}^{K-1} A'_{v_i} = 0$ (13)

- *Given the pairs* $\{A'_j, D'_j\}$ *fulfilling* (12) *and* (13) *and arbitrary pairs* $\{B'_j, C'_j\}$ *of* $\mathcal{D}$*, the system* $\mathcal{C}$ *reads*

$$\left\{\begin{array}{ll} x_{k+1} & = (A'_{\sigma(k)} - B'_{\sigma(k)}(D'_{\sigma(k)})^{-1}C'_{\sigma(k)})x_k \\ & \quad + B'_{\sigma(k)}(D'_{\sigma(k)})^{-1}u_k \\ y_k & = -(D'_{\sigma(k)})^{-1}C'_{\sigma(k)}x_k + (D'_{\sigma(k)})^{-1}u_k \end{array}\right. \tag{14}$$

Proof. The input u_k can be derived from the output equation of (14) and reads

$$u_k \quad = \quad D'_{\sigma(k)}y_k + C'_{\sigma(k)}x_k \tag{15}$$

Thus, from (11) and (15), one gets

$$\begin{aligned}\widehat{u}_k - u_k &= C'_{\sigma(k)}\widehat{x}_k + D'_{\sigma(k)}y_k - D'_{\sigma(k)}y_k - C'_{\sigma(k)}x_k \\ &= C'_{\sigma(k)}(\widehat{x}_k - x_k)\end{aligned}$$

Let the reconstruction error be $\epsilon_k = \widehat{x}_k - x_k$. Then, from (11), (14) and (15),

$$\begin{aligned}\epsilon_{k+1} &= A'_{\sigma(k)}\widehat{x}_k + B'_{\sigma(k)}y_k \\ &-(A'_{\sigma(k)} - B'_{\sigma(k)}(D'_{\sigma(k)})^{-1}C'_{\sigma(k)})x_k \\ &-B'_{\sigma(k)}(D'_{\sigma(k)})^{-1}u_k \\ &= A'_{\sigma(k)}\epsilon_k - B'_{\sigma(k)}(y_k - (D'_{\sigma(k)})^{-1}u_k) \\ &-B'_{\sigma(k)}(D'_{\sigma(k)})^{-1}u_k + B'_{\sigma(k)}y_k \\ &= A'_{\sigma(k)}\epsilon_k\end{aligned} \tag{16}$$

After iterating (16) K times and taking into account (13), one gets $\epsilon_k = 0$ or equivalently $x_k = \widehat{x}_k$ for any $k \geq K$. Hence, according to Definition 1, the set-up (10)–(11) is finite-time self-synchronizing. That completes the proof. □

Remark 1. *Condition* (13) *means that regardless of the order of multiplication of the matrices A'_j, and so for any mode sequences, the product is zero after a finite number K of iterations. K is the delay of synchronization.*

Remark 2. *The condition $D'_j \neq 0$ for any $j \in \mathcal{J}$ means that the relative degree of the systems* (11) *is zero. Such an assumption is mandatory so that* (11) *makes sense in the context of cryptography. Indeed, the deciphering must be a function of the state vector and the cryptogram (here the output of* (10)*)*

Remark 3. *The system* (10) *with the state space realization* (14) *is a right inverse for the system* (11)*. Indeed, for any identical initial conditions $x_0 = \widehat{x}_0$ and for any identical mode sequence $\{v\}$, the system* (14) *drives* (11) *such that $\forall k \geq 0$, $\widehat{u}_k = u_k$.*

The purpose of the next paragraph is to provide a constructive solution for the selection of appropriate matrices A'_j which must fulfill (13) in Theorem 1. It is based on the notion of nilpotent semigroups.

3.2. Nilpotent semigroups approach for finite-time self-synchronization

Let us first recall two definitions:

Definition 2 (Semigroup)**.** *A semigroup $\mathcal{S}$ is a set together with an associative internal law. It is said to be finite if $\mathcal{S}$ has a finite number of elements.*

Definition 3 (Nilpotent semigroup)**.** *A semigroup $\mathcal{S}$ is said to be nilpotent if any product of a finite number $t \in \mathbb{N}^*$ of its elements (possibly the same element) is always 0. The smallest integer t is called the class of nilpotency of $\mathcal{S}$.*

Proposition 1. *In order for* (13) *to be fulfilled, the set of dynamical matrices* $\{A'_j,\, j \in \mathcal{J}\}$ *must generate a nilpotent semigroup. The delay of synchronization* K *equals the class of nilpotency of this semigroup.*

A theorem, useful for the construction of semigroups with a given class of nilpotency is stated in the book [12] (Theorem 2.1.7) and recalled below.

Theorem 2 (Levitsky's theorem)**.** *Any semigroup of nilpotent matrices can be triangularized.*

In other words, all the matrices of a same nilpotent semigroup can be rewritten as upper triangular matrices up to a common linear transform (common basis). Since they are nilpotent their diagonal is zero.

Remark 4. *The product of* t *nilpotent matrices which commute pairwise is 0 but the product of* t *nilpotent matrices is not, in general, nilpotent. Indeed, we observe that* $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$.
Theorem 2 provides a generalization of this special case, should each matrix be nilpotent is only a necessary condition.

Hence, based on Levitsky's theorem, the construction of the family $\{A'_j,\, j \in \mathcal{J}\}$ which fulfills (13) follows three successive steps

Constructive Approach 1.

- *choose an invertible matrix* $T \in \mathbb{F}^{n\times n}$
- *choose a set of* J *upper triangular matrices* $\bar{A}'_j$ *with zero on the diagonal*
- *for all* $j \in \mathcal{J}$*, compute* $A'_j = T^{-1}\bar{A}'_j T$

The matrix T may clearly be the identity matrix.

Remark 5. *Because of Levitzky's theorem, the consideration of a semigroup of* n*–dimensional matrices is equivalent to the consideration of the corresponding set of upper triangular matrices. And yet, for triangular matrices, it is clear that the class of nilpotency is at most* n*. As a result, the delay of synchronization* K *is upper bounded by* n*.*

3.3. Connection between flat systems and the self-synchronizing canonical form

Flatness is an important concept in control theory. It was introduced by Fliess in [7] and a deep study can be found in the book [14]. In this section, we show that the constructive approach proposed for designing a finite-time self-synchronizing master-slave system amounts to designing a flat system $\mathcal{C}$ with flat output y_k and that the resulting SSSC can be written in the canonical form (1)–(2).

Definition 4 (Flat dynamical system [17])**.** *A system with input u_k and state vector x_k is said to be flat if there is a set of independent variables y_k, referred to as flat output, such that all the system variables can be expressed as a function of the flat output and a finite number of its backward and/or forward iterates. In particular, there exist two functions $\mathcal{F}$ and $\mathcal{G}$ such that*

$$\left\{ \begin{array}{ll} x_k & = \mathcal{F}(y_{k+k_1}, \ldots, y_{k+k_2}) \\ u_k & = \mathcal{G}(y_{k+k'_1}, \ldots, y_{k+k'_2}) \end{array} \right. \tag{17}$$

where $k_1, k_2, k'_1, k'_2 \in \mathbb{Z}$.

Proposition 2. *The system $\mathcal{C}$ resulting from the conditions* (12)-(13)-(14) *is flat with flat output y_k.*

Proof. The state of the switched system (11) can be written, at time $k+K$

$$\begin{array}{ll} \widehat{x}_{k+K} & = \prod_{i=0}^{K-1} A'_{\sigma(k+K-1-i)} \widehat{x}_k \\ & + \sum_{i=0}^{K-1} \left[\prod_{j=i+1}^{K-1} A'_{\sigma(k+K-j)} \right] B'_{\sigma(k+i)} y_{k+i} \end{array}$$

Therefore, if (13) holds, any state at time $k \geq 0$ reads:

$$\widehat{x}_{k+K} = \sum_{i=0}^{K-1} \left[\prod_{j=i+1}^{K-1} A'_{\sigma(k+K-j)} \right] B'_{\sigma(k+i)} y_{k+i} \tag{18}$$

And yet, according to the proof of Theorem 1, $\epsilon_k = 0$ or equivalently $x_k = \widehat{x}_k$ for any $k \geq K$. Hence, after a shift of K, one obtains

$$\widehat{x}_k = x_k = \sum_{i=0}^{K-1} \left[\prod_{j=i+1}^{K-1} A'_{\sigma(k-j)} \right] B'_{\sigma(k-K+i)} y_{k-K+i} \tag{19}$$

which gives the function $\mathcal{F}$.
On the other hand, the input u_k reads like (15). Substituting the expression (19) of x_k into (15) gives the function $\mathcal{G}$. That completes the proof. □

Hence, under the flatness condition, the systems (10)–(11) can be equivalently rewritten into the canonical form (1)–(2) and read

$$\left\{ \begin{array}{ll} x_k & = \sum_{i=0}^{K-1} \left[\prod_{j=i+1}^{K-1} A'_{\sigma(k-j)} \right] B'_{\sigma(k-K+i)} y_{k-K+i} \\ y_k & = C_{\sigma(k)} x_k + D_{\sigma(k)} u_k \end{array} \right. \tag{20}$$

$$\left\{ \begin{array}{ll} \widehat{x}_k & = \sum_{i=0}^{K-1} \left[\prod_{j=i+1}^{K-1} A'_{\sigma(k-j)} \right] B'_{\sigma(k-K+i)} y_{k-K+i} \\ \widehat{u}_k & = C'_{\sigma(k)} \widehat{x}_k + D'_{\sigma(k)} y_k \end{array} \right. \tag{21}$$

It is worth pointing out that, from a computational point of view, the recursive form (10)–(11) that is, the use of dynamical systems, is more relevant than (20)–(21).

4. Transmission zeros and surjectivity

For cryptographic purposes (basically a consideration regarding the entropy of sequences), it is relevant that the maps $x_k \mapsto A_j x_k$, $j \in \mathcal{J}$ are surjective. In other words, the following rank condition must be guaranteed:

$$\forall j \in \mathcal{J}, \ \operatorname{rank}(A_j) = n \tag{22}$$

The problem lies in that, according to Theorem 1, the matrices (A_j, B_j, C_j, D_j) of the system $\mathcal{C}$ are not designed directly but are derived from (A'_j, B'_j, C'_j, D'_j) of $\mathcal{D}$. Hence, we must find out a condition on the matrices (A'_j, B'_j, C'_j, D'_j) so that (22) is ensured. It turns out that the notion of *transmission zeros* is relevant to this end.

A definition of transmission zeros can be found for example in [13]. It is recalled below and particularized for a SISO system.

Definition 5. *Let us consider a SISO linear system with state space realization* (A, B, C, D). *The* transmission zeros *are the complex numbers* $\{s_i\}$ *which fulfill*

$$\operatorname{rank} \begin{bmatrix} A - s_i \mathbf{1}_n & B \\ C & D \end{bmatrix} < n + 1 \tag{23}$$

where it is recalled that $\mathbf{1}_n$ stands for the identity matrix of dimension n. The matrix involved in (23) is often called the Rosenbrock matrix.

Before proceeding further, let us introduce a few additional notations. Consider the invertible matrix T and the corresponding matrices $A'_j = T^{-1} \bar{A}'_j T$ derived from $\bar{A}'_j$ for $j \in \mathcal{J}$ as explained in Section 3.2 devoted to the Constructive Approach 1.

Let us write the upper triangular matrix $\bar{A}'_j$ as

$$\bar{A}'_j = \begin{bmatrix} 0 & a_j^1 & & & \\ & 0 & a_j^2 & A_j^* & \\ & & \vdots & \ddots & \\ & \mathbf{0} & & 0 & a_j^{n-1} \\ & & & \cdots & 0 \end{bmatrix} \tag{24}$$

where A_j^* denotes the coefficients above the $n-1$ diagonal entries a_j^m ($m = 1, \ldots, n-1$) located above the zero diagonal. Let have

$$TB'_j = [b_j^1 \cdots b_j^n]^T \tag{25}$$

b_j^m stands for the m^{th} component of the column vector TB'_j.

$$C'_j T^{-1} = [c_j^1 \cdots c_j^n] \tag{26}$$

c_j^m stands for the m^{th} component of the row vector $C_j' T^{-1}$.

Proposition 3. *The surjectivity of each map $x_k \mapsto A_j x_k$ ($j \in \mathcal{J}\}$) of $\mathcal{C}$ is guaranteed whenever*

$$c_j^1 b_j^n \prod_{m=1}^{n-1} a_j^m \neq 0 \tag{27}$$

Proof. According to Remark 3, (10) is a right inverse for (11). Furthermore, let us recall that (see Remark 2) the relative degree of (10) and (11) is zero. We conclude that each realization (A_j', B_j', C_j', D_j') ($j \in \mathcal{J}$) of $\mathcal{D}$ has n transmission zeros s_i and the s_i's are nothing but the n eigenvalues λ_i of A_j of (10). They are the roots of

$$\Psi_j(s) = \det\ R = 0 \ \text{ with } \ R = \begin{bmatrix} A_j' - s\mathbf{1}_n & B_j' \\ C_j' & D_j' \end{bmatrix} \tag{28}$$

where $\Psi_j(s)$ is a polynomial, its constant monomial is $\Psi_j(0)$ and corresponds to the product of the roots of $\Psi_j(s)$ and so corresponds to the product $\prod_{i=1}^n \lambda_i$ of the eigenvalues of A_j of $\mathcal{C}$. Hence, surjectivity of $x_k \mapsto A_j x_k$ $j \in \mathcal{J}$ is guaranteed whenever $\Psi_j(0) \neq 0$. The following equalities apply

$$\begin{aligned} \Psi_j(0) &= \det \begin{bmatrix} A_j' & B_j' \\ C_j' & D_j' \end{bmatrix} = \det \begin{bmatrix} T^{-1}\bar{A}_j' T & B_j' \\ C_j' & D_j' \end{bmatrix} \\ &= \det \left(\begin{bmatrix} T^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \bar{A}_j' & TB_j' \\ C_j' T^{-1} & D_j' \end{bmatrix} \begin{bmatrix} T & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \right) \\ &= \det \begin{bmatrix} \bar{A}_j' & TB_j' \\ C_j' T^{-1} & D_j' \end{bmatrix} \end{aligned} \tag{29}$$

Consider a partitioned matrix with four sub-blocks E, F, G, H of compatible dimensions. We recall a result concerning its determinant.

$$\det \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \det(H) \cdot \det(E - FH^{-1}G)$$

Hence, taking into account the special structure (24) of $\bar{A}_j'$, (25) and (26), basic manipulations yield

$$\Psi_j(0) = \det \begin{bmatrix} \bar{A}_j' & TB_j' \\ C_j' T^{-1} & D_j' \end{bmatrix} = c_j^1 b_j^n \prod_{m=1}^{n-1} a_j^m \tag{30}$$

That completes the proof. □

5. Statistical self-synchronization

So far, we have proposed a construction which guarantees self-synchronization with a finite delay K. This assumption limits the complexity of the ciphering

which can be represented as a memoryless function. This requirement is not mandatory in practice, and it is acceptable that the synchronization delay is not a constant value but a random variable with a probability law that reaches one as time reaches infinity. In this case, self-synchronization is said to be statistical [15]. Statistical self-synchronization is more general than the finite-time one. Its interest lies in a broader choice of candidate dynamical systems. The resulting flexibility is important in view of matching additional constraints, besides the self-synchronization, in particular regarding the security of the communication setup. It could be interesting to relax the finite-time synchronization constraint so that the synchronization probability follows a probability law that ensures synchronization for a large enough random sequence $\{y\}$. To this end, we propose, as an extension, to introduce the statistical self-synchronization.

Definition 6 (Statistical self-synchronization)**.** *The unidirectional coupled system $\mathcal{C}$–$\mathcal{D}$ is statistically self-synchronizing if*

$$\forall x_0,\, \widehat{x}_0,\, \lim_{k\to+\infty} \Pr[x_k = \widehat{x}_k] = 1 \tag{31}$$

where $\Pr[\cdot]$ stands for the probability, x_k and $\hat{x}_k$ being considered as random variables.
Let us stress that over the field of real numbers, we could have relaxed the finite-time synchronization constraint by allowing asymptotical synchronization with prescribed exponential decay rate. Over finite fields as it is the case here, asymptotical synchronization does no longer make sense.

5.1. General conditions

We give an equivalent theorem to Theorem 1 that corresponds to this situation. Note that the only difference with Theorem 1 concerns (13) which turns into (33).

Theorem 3. *The set-up* (10)–(11) *is statistically self-synchronizing whenever the three following conditions are fulfilled:*

- $\forall j \in \mathcal{J},\, D'_j \neq 0$ (32)

- $\exists K \in \mathbb{N},\, \forall x_0,\, \widehat{x}_0,\, \exists \{v\} \in \mathcal{J}^K,\, \prod_{i=0}^{K-1} A'_{v_i} = 0$ (33)

- *Given the pairs $\{A'_j, D'_j\}$ fulfilling* (32) *and* (33) *and arbitrary pairs $\{C'_j, D'_j\}$ of $\mathcal{D}$, the system $\mathcal{C}$ reads*

$$\begin{cases} x_{k+1} &= \left(A'_{\sigma(k)} - B'_{\sigma(k)}(D'_{\sigma(k)})^{-1}C'_{\sigma(k)}\right) x_k \\ &\quad + B'_{\sigma(k)}(D'_{\sigma(k)})^{-1}u_k \\ y_k &= -(D'_{\sigma(k)})^{-1}C'_{\sigma(k)}x_k + (D'_{\sigma(k)})^{-1}u_k \end{cases} \tag{34}$$

Proof. The proof follows the same development than the one of Theorem 1 till Equation (16). Equation (33) means that there exists a finite sequence of length K so that the product of K matrices A'_j is zero. Considering that any finite sequence has the probability one to appear in an infinite sequence (provided that any symbols has a non null probability of occurrence), satisfying (33) implies satisfying Equation (31) of Definition 6. □

Likewise the finite-time self-synchronization case, it should be interesting to check for constructive conditions, equivalent to (33), but with additionally the ability of controlling the probability of the synchronization delay while designing the system. Again, it turns out that we can resort to nilpotent semigroups.

5.2. Nilpotent semigroups for statistical self-synchronization

The proposed construction considers ℓ distinct nilpotent semigroups $\mathcal{S}_i$, $i \in \{1, \ldots, \ell\}$ of square n dimensional matrices each generated by a set of matrices $\mathcal{N}_i$. The cardinality of the set $\mathcal{N}_i$ ($i \in \{1, \ldots, \ell\}$) is denoted by J_i. The construction of the family $\{A'_j,\, j \in \mathcal{J}\}$ which fulfills (33) obeys the three following steps

Constructive Approach 2.

- *for each nilpotent semigroup $\mathcal{S}_i$ ($i \in \{1, \ldots, \ell\}$) to be built, choose a distinct invertible matrix $T_i \in \mathbb{F}^{n\times n}$*
- *for $i \in \{1, \ldots, \ell\}$, choose a set of J_i upper triangular matrices $\bar{A}'_{j'}$, $j' \in \{1, \ldots, J_i\}$ with zeros on the diagonal*
- *for $i \in \{1, \ldots, \ell\}$, for $j' \in \{1, \ldots, J_i\}$, compute $A'_j = T_i^{-1}\bar{A}'_{j'}T_i$ with $j = (i-1)\ell + j'$. These matrices are the elements of the set $\mathcal{N}_i$.*

Let t_i be the class of nilpotency of $\mathcal{S}_i$. The synchronization of $\mathcal{C}$–$\mathcal{D}$ is ensured if the switching rule σ selects t_i successive modes in the same nilpotent semigroup $\mathcal{S}_i$.

Remark 6. *It is worth emphasizing that finite-time synchronization is a special case of statistical self-synchronization which corresponds to $\ell = 1$, $J_1 = J$.*

5.3. Synchronization probability

When considering statistical self-synchronization, a question of interest is the shape of the synchronization probability function. Such a system is viable only if, for sequences of reasonable length, the synchronization probability is close to one. From the cryptographic point of view, it is also important that the synchronization does not occur too quickly. Such a system would be vulnerable to brute-force attacks that is, exhaustive search of the secret key.

The parameters that can be used to control the synchronization delay while designing the system are essentially, the dimension n of the system, the number ℓ of nilpotent semigroups $\mathcal{S}_i$, the number of generators J_i and the class of nilpotency t_i of $\mathcal{S}_i$. Section 6.2 illustrates such a purpose.

6. Illustrative examples

6.1. Finite-time self-synchronization

This section gives an example that illustrates the construction of a finite-time self-synchronizing setup. We propose to design a finite-time self-synchronizing system of dimension $n = 3$ and with $J = 3$ modes. The matrices are defined over the finite field $\mathbb{F} = \mathbb{Z}/7\mathbb{Z}$. Hence, their entries belong to the set of integers $\{0, \ldots, 6\}$ and the operations of additions and multiplications are performed modulo 7.

The design consists in a selection of matrices A'_j, B'_j, C'_j, D'_j which must fulfill the conditions of Theorem 1, the condition (13) being replaced by the Constructive Approach 1 provided in Section 3.2. The condition (27) on surjectivity is also incorporated into the constraints. Now let us detail the design.

First, for simplicity, we choose $D'_j = 1$ for any $j \in \mathcal{J} = \{1, 2, 3\}$ (condition (12)).
Secondly, following the Constructive Approach 1, it is chosen a set of three 3–dimensional matrices $\bar{A}'_j$ in the form of strict upper triangular matrices and with non zero entries located above the diagonal for the surjectivity (Condition (27)).

$$\bar{A}'_1 = \begin{pmatrix} 0 & 3 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \bar{A}'_2 = \begin{pmatrix} 0 & 2 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix} \bar{A}'_3 = \begin{pmatrix} 0 & 1 & 3 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}$$

Then, let us choose an invertible matrix T

$$T = \begin{pmatrix} 4 & 0 & 5 \\ 1 & 5 & 2 \\ 5 & 5 & 5 \end{pmatrix}$$

Its inverse over $\mathbb{F} = \mathbb{Z}/7\mathbb{Z}$ reads

$$T^{-1} = \begin{pmatrix} 4 & 2 & 5 \\ 6 & 1 & 2 \\ 4 & 4 & 3 \end{pmatrix}$$

Applying the change of basis $A'_j = T^{-1}\bar{A}'_j T$, we get that

$$A'_1 = \begin{pmatrix} 5 & 5 & 4 \\ 6 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} A'_2 = \begin{pmatrix} 6 & 3 & 0 \\ 3 & 2 & 1 \\ 5 & 2 & 6 \end{pmatrix} A'_3 = \begin{pmatrix} 0 & 2 & 4 \\ 1 & 4 & 0 \\ 6 & 1 & 3 \end{pmatrix}$$

Next, we choose arbitrary matrices B'_j and C'_j except the fact that the first entry c'^1_j of $C'_j T^{-1}$ and last entry b'^n_j of TB'_j are not zero to fulfil the surjectivity condition (27).

$$B'_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, B'_2 = \begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix}, B'_3 = \begin{pmatrix} 3 \\ 6 \\ 1 \end{pmatrix}$$

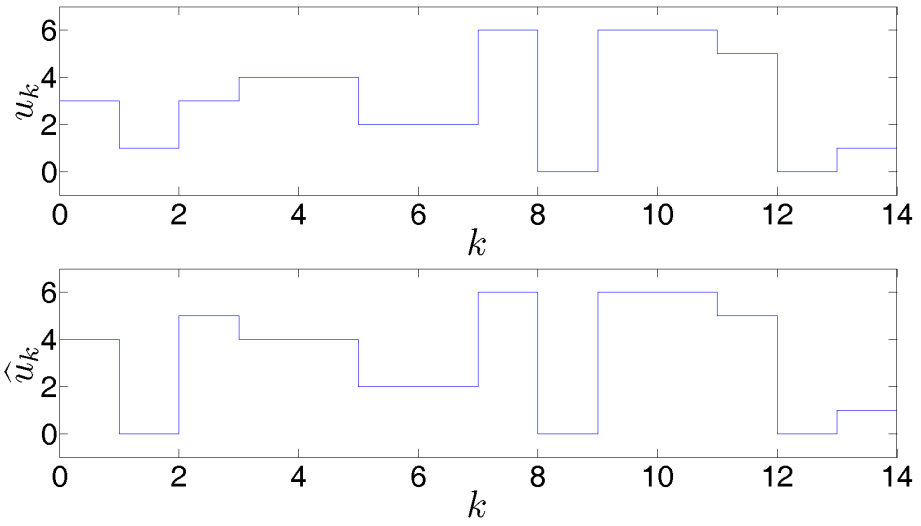

Figure 4: Time evolution of $\{u\}$ and $\{\widehat{u}\}$ of the setup $\mathcal{C}$-$\mathcal{D}$

$$C_1' = \begin{pmatrix} 2 & 1 & 3 \end{pmatrix}, C_2' = \begin{pmatrix} 6 & 2 & 1 \end{pmatrix}, C_3' = \begin{pmatrix} 3 & 1 & 1 \end{pmatrix}$$

Finally, we derive the equations (14) of the right inverse system $\mathcal{C}$. The matrices read

$$A_1 = \begin{pmatrix} 6 & 5 & 4 \\ 6 & 1 & 3 \\ 0 & 0 & 4 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 1 & 6 \\ 5 & 5 & 6 \\ 3 & 6 & 1 \end{pmatrix}, A_3 = \begin{pmatrix} 5 & 6 & 1 \\ 4 & 5 & 1 \\ 3 & 0 & 2 \end{pmatrix}$$

$$B_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, B_2 = \begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix}, B_3 = \begin{pmatrix} 3 \\ 6 \\ 1 \end{pmatrix}$$

$$C_1 = \begin{pmatrix} 5 & 6 & 4 \end{pmatrix}, C_2 = \begin{pmatrix} 1 & 5 & 6 \end{pmatrix}, C_3 = \begin{pmatrix} 4 & 6 & 6 \end{pmatrix}$$

$$D_1 = D_2 = D_3 = 1$$

Having completed the design of (10)–(11), a sequence $\{u\}$ is applied to $\mathcal{C}$. As expected, the self-synchronization is achieved after a finite transient time, and so does the recovery of the sequence of inputs (see Figure 4). The transient time before self-synchronization lasts $K = 3$ samples. It is in accordance with the class of nilpotency $t = 3$ of the set $\{A_1', A_2', A_3'\}$.

6.2. Statistical self-synchronization

In this example we aim at designing a setup (10)–(11) having the statistical self-synchronization property. Besides, we assess the impact of the variation of the number of nilpotent semigroups ℓ on the synchronization delay. The dimension of the dynamical system is $n = 10$. The number of nilpotent semigroups

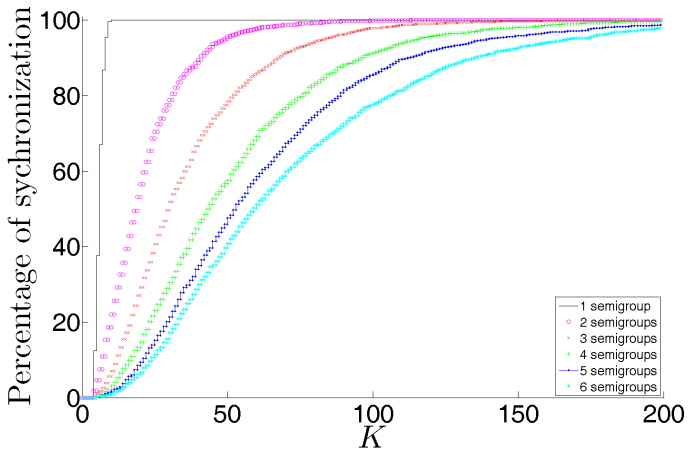

Figure 5: Percentage of the number of times the system $\mathcal{C}$–$\mathcal{D}$ has synchronized with respect to the delay of synchronization K for different number ℓ of nilpotent semigroups

varies from $\ell = 1$ to 6. They are built according to the Constructive Approach 2 given in Section 5.2. The nilpotent semigroups have the same number of generators $J_i = 5$ and class of nilpotency $t_i = 10$ $\forall i \in \{1, \ldots, \ell\}$. The experiment is conducted by generating random mode sequences $\{v\} = \{\sigma(0), \sigma(1), \ldots\}$ and determining, over 2000 runs, the percentage of sequences for which self-synchronization occurs. The result is depicted in Figure 5. The experiment shows that the more semigroups the higher the synchronization delay on average. In any case, the percentage gets close to 100% as K increases and is in accordance with Definition 6. The case when there is only one nilpotent semigroup deserves a special comment. The curve reaches 100% after $K = 10$. Indeed it corresponds to a finite self-synchronization according to Remark 6. The delay $K = 10$ corresponds to the class of nilpotency $t = 10$ of the set of matrices A'_j.

The probability law of synchronization seems to have an exponential-like shape. It is not trivial to figure out the exact expression of the law. Indeed, the problem amounts to determine the probability of occurrence of the mode sequences $\{v\}$ that induce self-synchronization. And yet, it is shown in [16] that a general treatment of this issue can be very intricate.

7. Conclusion

This paper has addressed the problem of self-synchronization with potential applications to secure communications. The setup under considerations involved two dynamical systems coupled in a unidirectional way. Two kinds of self-synchronization have been investigated: the finite-time one and, as an extension, the statistical one. For discrete-time switched linear systems, it has been shown that a variety of concepts borrowed from control theory, namely

flatness, right invertibility, transmission zeros are relevant to design an admissible setup. It turns out that the notion of nilpotent semigroups is central to guarantee the finite-time self-synchronization. More refined characterization of the probability law defining the delay of synchronization in the statistical case should be examined in a near future. The control theoretical concepts addressed through this work can be considered as a first step towards a complete framework for designing self-synchronizing stream ciphers. Clearly, security criteria must be further incorporated into this framework. They have not been discussed here because out of the scope but can be found in companion papers.

Acknowledgement

This work was supported in part by the Institut des Sciences et de l'Ingénierie des Systèmes, Centre National de la Recherche Scientifique, Programmes Exploratoires Pluridisciplinaires (PEPS), Projet Autocrypt.

References

[1] S. Banerjee, editor. *Chaos Synchronization and Cryptography for Secure Communications: Applications for Encryption*. IGI Global, 2010.

[2] I.I. Blekhman, A.L. Fradkov, Nijmeijer H., and A.Y Pogromsky. On self-synchronization and controlled synchronization. *Systems and Control letters*, 31(5):299–305, 1997.

[3] C. Carlet. *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, chap. Vectorial Boolean Functions for Cryptography. Cambridge Press, 2010.

[4] F. Dachselt and K. Kelber and J. Vandewalle and W. Schwarz. *Chaotic versus classical stream ciphers – A comparative study Proc. of Int. Symp. on Circuits and Systems ISCAS'98*,IV518–521, ,1998

[5] J. Daemen. *Cipher and Hash function design, strategies based on linear and differential cryptanalysis*. PhD Thesis, Katholieke Univ. Leuven, 1995.

[6] J. Daemen and P. Kitsos. The self-synchronizing stream cipher moustique. *eSTREAM, ECRYPT Stream Cipher Project*, June 2005. Available online at http://www.ecrypt.eu.org/stream.

[7] M. Fliess, J. Levine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *Int. J. of Control*, 61(6):1327–1361, 1995.

[8] U. M. Maurer. New approaches to the design of self-synchronizing stream cipher. *Advance in Cryptography, In Proc. Eurocrypt '91, Lecture Notes in Computer Science*, pages 548–471, 1991.

[9] G. Millérioux and P. Guillot. Self-synchronizing stream ciphers and dynamical systems: state of the art and open issues. *International Journal of Bifurcation and Chaos*, 20(9), September 2010.

[10] G. Millérioux, J. M. Amigó, and J. Daafouz. A connection between chaotic and conventional cryptography. *IEEE Trans. on Circuits and Systems I: Regular Papers*, 55(6), July 2008.

[11] J. Parriaux, P. Guillot, and G. Millérioux. Synchronization of boolean dynamical systems: a spectral characterization. In Claude Carlet and Alexander Pott, editors, *Sequences and Their Applications (SETA 2010)*, volume 6338 of *Lecture Notes in Computer Science*, Paris, France, September 2010. Springer Berlin / Heidelberg.

[12] H. Radjavi and P. Rosenthal. *Simultaneous Triangularization.* Springer, 2000.

[13] C. B. Schrader and M. K. Sain. Research on system zeros: a survey. *Int. Jour. of Control*, 50(4):1407–1433, 1989.

[14] H. Sira-Ramirez and S. K. Agrawal. *Differentially Flat Systems.* Marcel Dekker, New York, 2004.

[15] J. Parriaux and P. Guillot and G. Millérioux *Towards a spectral approach for the design of self-synchronizing stream ciphers.* Cryptography and Communications, Springer New York, 2011.

[16] D. Bajić, and C. Stefanović, *Statistical Analysis of Search for Set of Sequences in Random and Framed Data.* Sequences and Their Applications (SETA 2010), Springer Berlin.

[17] G. Millérioux and J. Daafouz *Flatness of switched linear discrete-time systems.* IEEE Trans. on Automatic Control, March 2009, Vol 54