# A connection between chaotic and conventional cryptography

Gilles Millérioux, Jose Maria Amigo, Jamal Daafouz

# A connection between chaotic and conventional cryptography

Gilles Millérioux, José Maria Amigó, Jamal Daafouz

***Abstract*— A lot of encryption methods involving chaotic dynamics have been proposed in the literature since the 90's. Most of them consists of "mixing" the confidential information being transmitted through an insecure channel, with a chaotic analog or digital sequence. The recovering of the original information usually calls for reproducing, at the receiver side, the same chaotic signal as at the transmitter side. The synchronization mechanism of the two chaotic signals is known as *chaos synchronization*. In this paper, a connection between chaotic and conventional encryption is established with special emphasis on two of the most attractive schemes, namely, *message-embedding* and *hybrid message-embedding*. The main point of this survey can be stated as follows: (hybrid) message-embedding is strictly equivalent to a conventional self-synchronizing stream cipher under *flatness conditions*.**

## I. Introduction

Nowadays information is electronically processed and conveyed through public networks. The main objective of cryptography is, precisely, to conceal the content of messages transmitted through insecure channels to unauthorized users or, in other words, to guarantee privacy and confidentiality in the communications. Since the early 1960s, cryptography has no longer been restricted to military or governmental concerns, what has spurred an unprecedented development of it. At the same time, this development benefited very much from the advances in digital communication technology in form of new and efficient ways of designing encryption schemes. Let us shortly recall that modern cryptography originates in the works of Feistel at IBM during the late 1960s and early 1970s. One of the key dates is 1977, when the Data Encryption Standard (DES) was adopted by the U.S. National Bureau of Standards (now the National Institute of Standards and Technology — NIST), for encrypting unclassified information. DES is now in the process of being replaced by the Advanced Encryption Standard (AES), a new standard adopted by NIST in 2001. Another milestone is 1978, marked by the publication of RSA, the first full-fledged public-key algorithm. This discovery not only solved the key-exchange problem of symmetric (or private-key) cryptography but, most importantly, did it open new whole areas (like authentication and electronic signature) in modern cryptology.

Gilles Milll´erioux is with the University Henri Poincar´e of Nancy at the Centre de Recherche en Automatique de Nancy in France (UHP-CRAN) Email:gilles.millerioux@esstin.uhp-nancy.fr

Jos´e Maria Amig´o is with the University Miguel Hernandez of Elche (UMH) at the Centro de Investigaci´on Operativa in Spain Email:jm.amigo@umh.es

Jamal Daafouz is with the Institut National Polytechnique de Lorraine at the Centre de Recherche en Automatique de Nancy in France (INPL-CRAN) Email:jamal.daafouz@ensem.inpl-nancy.fr

In 1993 entered the scene "chaotic cryptography", that takes advantage of the complex behavior of chaotic dynamical systems to 'hide' or 'mask' information. Since then, many different implementations of this basic idea have been proposed in the open literature. Chaotic behavior can be distinguished by its extreme sensitivity to initial conditions, leading to long-term unpredictability. Moreover, signals resulting from chaotic dynamics are broadband and present random-like statistical properties, albeit they are generated by deterministic systems. All this explains why there is likely a connection between the random-looking behavior exhibited by chaotic systems and the properties of confusion and diffusion, required by Shannon for cryptosystems [27]. It also motivates the use of chaotic systems for secure communications, even though the terminology "secure" is sometimes questionable. An overview of the different methods devised so far can be found, according to the chronology, in the papers [35][17][43]. Nevertheless, very few works have really established the connection between standard and chaos-based encryption algorithms, but see [8][21] for some interesting exceptions.

This paper contributes to a deeper insight in this issue by comparing the structures involved in chaotic and conventional cryptographic schemes, with a special treatment of symmetric encryption.

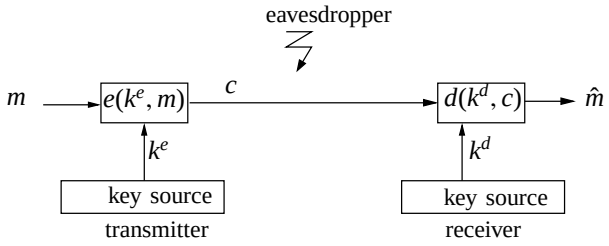


Fig. 1. General encryption mechanism

A general encryption mechanism or scheme, also called cryptosystem or cipher, is illustrated in Fig. 1. We are given an alphabet $\mathcal{A}$, that is, a finite set of basic elements named symbols. On the *transmitter* part, a plaintext (also called information or message) $m \in \mathcal{M}$ (the message space) consisting of a string of symbols $m_k \in \mathcal{A}$ is encrypted according to an encryption function $e$ which depends on the key $k^e \in \mathcal{K}$ (the key space). The resulting ciphertext $c \in \mathcal{C}$ (the ciphertext space), a string of symbols $c_k$ from an alphabet usually identical to $\mathcal{A}$, is conveyed through a channel to the *receiver*. At the receiver side, the ciphertext $c$ is decrypted according to a decryption function $d$ which depends on the key $k^d \in \mathcal{K}$. The function $e$ (*resp.* $d$) must

be a bijection from $\mathcal{M}$ to $C$ (*resp.* $C$ to $\mathcal{M}$ ). The encryption scheme corresponding to the pair $(e, d)$ must be designed in an appropriate way so as it is a hard task for an eavesdropper to retrieve the plaintext $m$. Therefore, there must exist a unique pair $(k^e, k^d)$ such that $d(k^d, c) = m$ where $c = e(k^e, m)$. Let us point out that the design of a cryptographic scheme must take into account that the sets $\mathcal{M}$, $C$, $\mathcal{K}$ and the pair $(e, d)$ are known. Only the pair $(k^e, k^d)$ can be assumed to be secret. This is a fundamental premise in cryptanalysis, first stated by A. Kerkhoff in 1883. As a matter of fact, in some special situations like public-key cryptography, only $k^d$ must be kept secret.

This paper consists of two parts. Section II reviews the most popular chaotic cryptosystems proposed over the years since 1993. The main respective advantages and drawbacks are also pointed out. Section III is devoted to the comparative study between these chaotic cryptosystems and the conventional symmetric ciphers, specifically, stream ciphers. In particular, it is shown there that (hybrid) message-embedded chaotic ciphers and conventional self-synchronizing stream ciphers are equivalent under the so-called flatness condition, a condition borrowed from control theory.

## II. Chaotic cryptosystems

There are basically two classes of chaotic cryptosystems. The first one amounts to numerically computing a great number of iterations of a discrete chaotic system, using the message as initial data (see [11][38] and references therein). This is basically also the strategy in [39][1], where periodic approximations of chaotic automorphisms are used to define substitutions (so-called S-boxes) resistant to linear and differential cryptanalysis. The second class, on which we shall actually focus in this paper, amounts to scrambling a message $m_k$ with a chaotic dynamic $f$. $f$ is often specified by a state representation with corresponding state vector $x_k$, the dimension of the system being $n$. $f$ is parametrized by a vector $\theta$ of dimension $L$ ($\theta = [\theta^{(1)}, \ldots, \theta^{(L)}]$) which is intended to act as the secret key. Only a part of the state vector $x_k$ obtained via a function $h$, possibly parametrized by $\theta$ as well, called the "output" and denoted by $y_k$, is conveyed through the public channel. $y_k$ is usually of low dimension and should be unidimensional in the ideal case. In what follows, we will assume that $y_k$ is a scalar (dimension 1), the transmitter being thus restricted to a so-called Single Input Single Output (SISO) system. The receiver is a dynamical system $\tilde{f}$, parametrized by $\hat{\theta}$ with output function $\tilde{h}_{\hat{\theta}}$. Throughout the paper, unless otherwise stated, it will be assumed that $\hat{\theta} = \theta$.

Usually, retrieving the message, that is, achieving $\hat{m}_k = m_k$ is performed in two steps.

The first step is called *synchronization*. It is based on a suitable choice of $\tilde{f}$ so that

$$\lim_{k\to\infty} \|T x_k - \hat{x}_k\| = 0 \quad \forall \hat{x}_0 \in U \tag{1}$$

or

$$\exists k_f < \infty, \quad \|T x_k - \hat{x}_k\| = 0 \quad \forall \hat{x}_0 \in U \quad \text{and} \quad k \geq k_f \tag{2}$$

where $T$ is a constant matrix of appropriate dimension and $U$ is a non empty set of initial conditions. (1) corresponds to an asymptotic synchronization, while (2) corresponds to a finite time synchronization. Let us point out that in practice, since we deal with finite accuracy, the error of an asymptotical synchronization can be considered to be zero after a finite transient time. As a matter of fact, synchronization can be viewed as a state reconstruction. In 1997 several papers [16][26][29][13] brought out this connection. As a result, the receiver often consists in an observer. If only a part of the components are reconstructed, the observer is a reduced observer and $rank(T) < n$. If all the components of the state vector are reconstructed, the observer is a full observer and $T$ is the identity matrix.

The second step consists in estimating $m_k$ through a suitable static function which depends on the internal state $\hat{x}_k$ and the output $y_k$.

Various cryptosystems, corresponding to distinct ways of hiding a message, have drawn the attention of the researchers over the years. They are reviewed in the following subsections. Let us point out that we are going to restrict to discrete-time systems (maps), but most of these chaotic cryptosystems can also be found in the literature for the continuous time.

### A. Additive masking

This scheme was first suggested in [24] and [42]. The information $m_k$ to be hidden is merely added to the output $y_k$ of the *transmitter* (Fig. 2):

$$\left\{ \begin{array}{lcl} x_{k+1} & = & f_\theta(x_k) \\ y_k & = & h_\theta(x_k) + m_k \end{array} \right. . \tag{3}$$

The generic equations of the *receiver* read:

$$\left\{ \begin{array}{lcl} \hat{x}_{k+1} & = & \tilde{f}_\theta(\hat{x}_k, y_k) \\ \hat{y}_k & = & \tilde{h}_\theta(\hat{x}_k) \end{array} \right. . \tag{4}$$

The quantity $y_k$ which appears in (4) reveals the unidirectional coupling between both the transmitter and the receiver systems. Provided that synchronization (1) or (2) can be achieved, the recovering of the information is performed by

$$\hat{m}_k = y_k - \hat{y}_k.$$

Unfortunately, usually the information cannot be exactly retrieved. Indeed, $m_k$ acts as a disturbance on the channel and precludes the *receiver* from being exactly synchronized; neither (1) nor (2) can be exactly fulfilled. As a result, $\hat{x}_k \neq x_k$, $\hat{y}_k \neq y_k$ and, finally, $\hat{m}_k \neq m_k$ for any $k$.

### B. Modulation

*1) Chaotic switching:* Chaotic switching is also referred to as chaotic modulation or chaos shift keying. Such a technique has been mostly proposed in the digital communications context. A description with deep insights can be found in

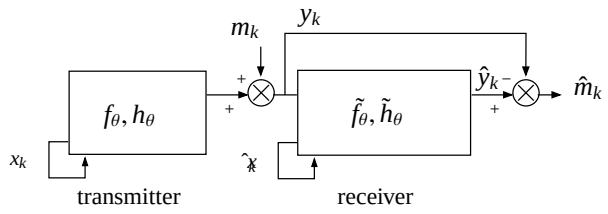


Fig. 2. Additive masking

[12], even though the method was proposed a couple of years before, say, in 1993 [15]. Basically, at the *transmitter*, to each symbol $m_k = m^i$ belonging to a finite set $\{m^1, \ldots, m^N\}$, it is assigned a chaotic signal emanating from the dynamic $f_\theta^i$ with output function $h_\theta^i$ ($i = 1, \ldots, N$). Therefore, in the transmitter description, the index $i$ depends on $m_k$.

$$\left\{ \begin{array}{rcl} x_{k+1} & = & f_\theta^{i(m_k)}(x_k) \\ y_k & = & h_\theta^{i(m_k)}(x_k) \end{array} \right. . \tag{5}$$

The simplest case involves binary-valued information and only two different chaotic dynamics $f^1$, $f^2$ are needed. Then, according to the current value of the symbol $m_k$ at times $k = jK$ ($j \in \mathbb{N}$), a switch is periodically triggered on every $K$ samples. During the interval of time $[jK, (j+1)K - 1]$, $m_k$ is assumed to be constant and the chaotic signal $y_k$ of the system which has been switched on is conveyed through the channel (Fig. 3).

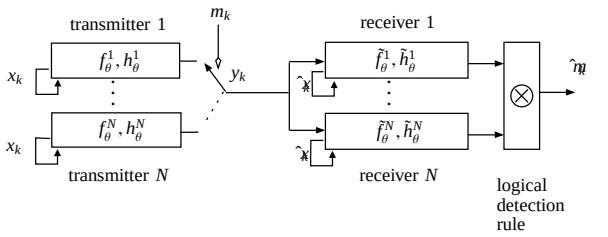


Fig. 3. Chaotic switching

The objective at the *receiver* end is to decide which chaotic system $f_\theta^i$ is most likely to have produced the sequence $\{y_k\}_{jK,\ldots,(j+1)K-1}$. To this end, the receiver part is composed of as many systems, say $N$, as at the transmitter part:

$$\left\{ \begin{array}{rcl} \hat{x}_{k+1} & = & \tilde{f}_\theta^i(\hat{x}_k, [y_k]) \\ \hat{y}_k & = & \tilde{h}_\theta^i(\hat{x}_k) \end{array} \right. \tag{6}$$

The symbol $[\cdot]$ distinguished two methods: the coherent and the non coherent detections. Non coherent detection involves statistical approaches mainly based upon correlation operations between the transmitted signal $y_k$ and the estimated signal $\hat{y}_k$. In this case, the receivers are autonomous systems with dynamics $\tilde{f}_\theta^i$ and $y_k$ must be omitted in (6). Coherent methods require the synchronization of both the transmitter and the receiver. The synchronization (1) or (2) (where $x_0$ must be replaced by $x_{jK}$) is obtained by unidirectional coupling through the variable $y_k$ which is really involved in $\tilde{f}_\theta^i$ of Eq. (6). Only one of the $N$ receivers (observers for instance) can be synchronized according to the value of $m_k$ which is constant within the interval of time $[jK, (j+1)K-1]$. A simple logical decoder permits to retrieve the original information when analyzing the residuals $r_k^i$, where

$$r_k^i = h_\theta^{i(m_k)}(x_k) - \tilde{h}_\theta^i(\hat{x}_k).$$

When multi-valued information is considered [36], the number of receivers increases and a more sophisticated logical mechanism, located after the bank of receivers, is required.

Regarding a noisy context, the modulation technique is appealing because it benefits from some immunity properties. In a noise-free context though, it is much less attractive because it suffers from the fact that each switch of $m_k$ causes a transient in the synchronization process. That motivates the requirement that $m_k$ must be constant within an interval of time. Unfortunately, that prevents from high throughput transmissions. Besides, the lack of effectiveness is reinforced by the number of receivers which may become huge when $N$ grows up.

*2) Parameter modulation:* Basically, two kinds of modulations can be distinguished: the discrete and the continuous one.

The setup corresponding to a discrete parameter modulation [41][15] is depicted in Fig. 4a. In such a case, a parameter $\lambda$ (different from the key parameter $\theta$) of a single chaotic system, takes values $\lambda(m_k) = \lambda^i$ according to a prescribed rule over a finite set $\{\lambda^1, \ldots, \lambda^N\}$. For binary messages, the parameter of the *transmitter* only takes two distinct values $\lambda^1$, $\lambda^2$. During the interval of time $[jK, (j+1)K - 1]$, $m_k$ is assumed to be constant and the chaotic signal $y_k$ is conveyed through the channel:

$$\left\{ \begin{array}{rcl} x_{k+1} & = & f_\theta^{\lambda(m_k)}(x_k) \\ y_k & = & h_\theta^{\lambda(m_k)}(x_k) \end{array} \right. . \tag{7}$$

The receiver part usually consists of a bank of $N$ receivers, usually some observers, each of them being coupled in a unidirectional way with the transmitter through $y_k$:

$$\left\{ \begin{array}{rcl} \hat{x}_{k+1} & = & \tilde{f}_\theta^{\lambda^i}(\hat{x}_k, y_k) \\ \hat{y}_k & = & \tilde{h}_\theta^{\lambda^i}(\hat{x}_k) \end{array} \right. . \tag{8}$$

Only one observer, set with the same value $\lambda^i$ of the transmitter which has actually delivered the sequence $\{y_k\}_{jK,\ldots,(j+1)K-1}$, can be synchronized in the form (1) or (2) (where $x_0$ must be replaced by $x_{jK}$) within the time interval $[jK, (j+1)K - 1]$. Thus, again, a simple logical decoder permits to retrieve the original information when analyzing the residuals

$$r_k^i = h_\theta^{\lambda(m_k)}(x_k) - \tilde{h}_\theta^{\lambda^i}(\hat{x}_k).$$

For the continuous modulation (Fig. 4b), the information $m_k$ takes values over an uncountable set $\mathcal{M}$. Consequently, an infinite number of units at the receiver side would be required. As a matter of fact, for the recovering of $\lambda_k$ and of $m_k$, we usually resort to adaptive techniques and identification procedures [10][6][14][2]. The estimation must fulfill $\hat{\lambda}_k = \lambda_k$ after a transient as short as possible.

The rule $\lambda(m_k)$ must be bijective so that performing $\lambda^{-1}(\lambda^i)$ for the discrete modulation or $\lambda^{-1}(\hat{\lambda}_k)$ for the continuous modulation, leads to a unique value $m_k$.

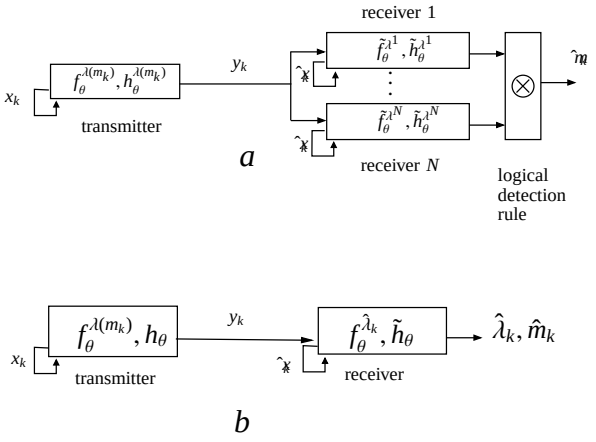


Fig. 4. Parameter modulation

Nevertheless, for the parameter modulation and similarly to the chaotic switching, the information must be constant during a prescribed interval of time (or at least slowly time-varying in a bounded range) to cope with the transients induced by the adaptive or the identification process. As happened with chaotic switching, this technique severely limits high throughput purposes and, therefore, it seems to be not very appealing for pure encryption perspectives.

### C. Message-embedding

*1) Structure:* First of all, the reader is cautioned that different but equivalent terminologies can be encountered in the literature referring to the same technique: embedding [19][31], non autonomous modulation [43] or direct chaotic modulation [17]. The reasons are the following. At the transmitter part, the information $m_k$ is directly injected (or, as it is also usually said, embedded) in a chaotic dynamic $f_\theta$. The resulting system turns into a non autonomous one since the information acts as an exogenous input. Injecting $m_k$ into the dynamic could be considered as a "modulation" of the phase space. Only the output $y_k$ of the system is transmitted. There exists two classes of systems. The first one corresponds to systems governed by the state equations

$$\begin{cases} x_{k+1} = f_\theta(x_k, m_k) \\ y_k = h_\theta(x_k, m_k) \end{cases}, \tag{9}$$

while the second class corresponds to

$$\begin{cases} x_{k+1} = f_\theta(x_k, m_k) \\ y_k = h'_\theta(x_k) \end{cases}. \tag{10}$$

The systems (9) and (10) differ from each other by their *relative degree*.

*Definition 1:* ([18] P.139) The relative degree of a system with respect to the quantity $m_k$ is the required number $r$ of iterations of the output $y_k$ so as $y_{k+r}$ depends on $m_k$ which actually appears explicitly in the expression of $y_{k+r}$.

*Remark 1:* For Single Input Single Output (SISO) linear systems, the relative degree $r$ corresponds to the difference between the degree of the denominator and the degree of the numerator in their transfer function

Based on Definition 1, the relative degree of the system (9) is $r = 0$.

The system (10) has a relative degree $r$ strictly greater than 0. It means that, after iterating $r$ times the state vector $x_k$, the output $y_{k+r}$ reads

$$y_{k+r} = h'_\theta(f^r_\theta(x_k, m_k)) \tag{11}$$

where

$$\begin{aligned} f^i_\theta(x_k, m_k) &= x_k \quad when\ i = 0 \\ &= f_\theta(f^{i-1}_\theta(x_k, m_k), m_{k+i-1}) \quad \forall i \geq 1. \end{aligned}$$

and where $m_k$ appears explicitly in the sense that there exists $m'_k \neq m_k$ such that $y_{k+r} = h'_\theta(f^r_\theta(x_k, m_k)) \neq h'_\theta(f^r_\theta(x_k, m'_k))$ whereas for all $m'_k \neq m_k$, $y_{k+r'} = h'_\theta(f^{r'}_\theta(x_k, m_k)) = h'_\theta(f^{r'}_\theta(x_k, m'_k))$ if $r' < r$.

Similarly to the previous techniques, the recovering of the message is performed in two steps. At the receiver side, $x_k$ must be first reconstructed by a synchronization mechanism in such a way that (1) or (2) is fulfilled. Then, $m_k$ can, in turn, be easily deduced. Two mechanisms have been proposed in the literature: the inverse system approach [40] and the unknown input observer approach [9][5][31][30][32][4]. The transmitter exhibits an output behavior that depends both on the internal chaotic state vector $x_k$ and on the input signal $m_k$. The role of the receiver is to reproduce the input $m_k$ given the only available data $y_k$ (and possibly their iterates). Hence, it acts as an inverse system. The main problem arising in the inverse approach proposed in [40] stems from the fact that the inverse system is likely to be unstable or to get bad performance properties in a noisy context. In such a case, this drawback must be redressed and a refinement of the design is needed. This leads naturally to some structures named Unknown Input Observers (UIO). As a matter of fact, UIO are nothing else but inverse systems slightly modified by adding some extra terms for the sake of convergence properties. The generic equations governing an inverse system or an UIO for (9) or (10) are, according to their relative degree $r \geq 0$:

$$\begin{cases} \hat{x}_{k+1} = \tilde{f}_\theta(\hat{x}_k, y_k, \ldots, y_{k+r}) \\ \hat{m}_k = g(\hat{x}_k, y_{k+r}) \end{cases}, \tag{12}$$

with $g$ such that

$$\hat{m}_k = g(\hat{x}_k, y_{k+r}) = m_k \quad \text{when} \quad \hat{x}_k = x_k. \tag{13}$$

The message-embedding is very attractive insofar as the synchronization (1) or (2) can be guaranteed without any restriction on the rate of variation of $m_k$.

*2) Security:* In this subsection we address the problem of the security of message-embedded cryptosystems. This is an essential issue for the validation of cryptosystems. Questions regarding security belong to cryptanalysis, that is the science of studying attacks against cryptographic schemes in order to reveal their possible weaknesses. As already mentioned

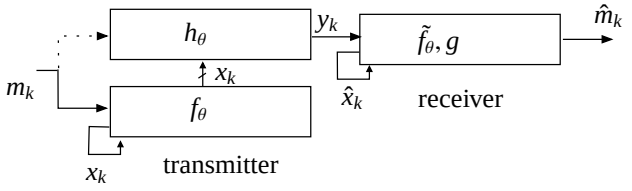


Fig. 5. Message-embedding. When $r = 0$, $m_k$ is embedded into $f_\theta$ and $h_\theta$. When $r > 0$, $m_k$ is only embedded into $f_\theta$

in the introduction, a fundamental premise in cryptanalysis that goes back to Kerkhoff [28], is that the adversary or the intruder knows all the details of the cryptosystem, including the algorithm and its implementation, except the secret key, on which the security of a cryptosystem must entirely rely. Hence, the (in general, vectorial) parameter $\theta$ of a chaotic cryptosystem plays a central role because it is intended to act as the secret key, and assessing the difficulty of its recovering is of paramount importance. Amazingly, only a somewhat limited interest has been attached so far to this fundamental issue in the case of chaotic ciphers, with the result that most of them (including additive masking and chaotic modulation) were proved to be insecure shortly after proposal. Now researchers in chaotic cryptography are becoming more aware of the necessity of cryptanalysis. Along these lines, we sum up next some results provided in the paper [3] for message-embedding.

Let us assume that, in order to retrieve the secret key, the eavesdropper tries exhaustively every possible value $\theta$ of the parameter space that, in practice, is a finite space. This procedure corresponds to the so-called brute force attack. The quicker the brute force attack succeeds, the weaker the cryptosystem is. Hence, the worst situation for the eavesdropper and the best from the viewpoint of security, is that there exists a unique parameter $\theta$ which induces a prescribed behavior at the output (by assumption, accessible to him). Indeed, the probability of finding the actual value is the lowest in this case. A key idea is that uniqueness is directly linked to parametric identifiability.

Some formal definitions of parametric identifiability are recalled below. Definition 3 and 4 are analytical definitions, based on the notion of *admissible input*.

*Definition 2:* An input sequence over a window of iterations $[0 - T]$, denoted by $\{m_k\}_0^T$, is called an *admissible input on* $[0 - T]$ if the difference equation (9) (or (10)) admits a unique local solution.

*Definition 3:* The system (9) (or (10)) is *locally strongly* $x_0$*-identifiable* at $\theta$ through the admissible input sequence $\{m_k\}_0^T$ if there exists an open neighborhood of $\theta$, $v(\theta) \subset \Theta$, such that for any $\hat{\theta} \in v(\theta)$ and for any $\theta \in v(\theta)$:

$$\hat{\theta} \neq \theta \quad \Rightarrow \quad \{y_k(x_0, m_k, \hat{\theta})\}_0^T \neq \{y_k(x_0, m_k, \theta)\}_0^T \tag{14}$$

where $\{y_k(x_0, m_k, \theta)\}_0^T$ (*resp.* $\{y_k(x_0, m_k, \hat{\theta})\}_0^T$) stands for the discrete trajectory $\{y_k\}_{0,\ldots,T}$ of the system parametrized by $\theta$ (*resp.* $\hat{\theta}$), initialized at $x_0$ and induced by $m_k$.

*Definition 4:* The systems (9) (*resp.* (10)) is *structurally identifiable* if there exist $T > 0$, an open subset $\mathcal{X}_0 \subset \mathcal{X}$ and some dense subsets $v(\theta) \subset \Theta$ and $\mathcal{M}_0^T \subset \mathcal{M}$, such that the system (9) (*resp.* (10)) is locally strongly $x_0$-identifiable at $\theta$ through the admissible input sequence $\{m_k\}_0^T$ for every $x_0 \in \mathcal{X}_0$, $\theta \in v(\theta)$ and $\{m_k\}_0^T \in \mathcal{M}_0^T$.

In the following, we will equivalently say that the system or its parameters are structurally identifiable.

Since identifiability is based upon input/output considerations, we will be able to conclude on the identifiability of $\theta$ if the state vector $x_k$ can be somehow eliminated from (9) (or (10)) to obtain an *input/output relation* of the form

$$\Phi(m_k, y_k; \theta) = 0, \tag{15}$$

which may also involve subsequent iterates of $m_k$ and $y_k$. For systems having polynomial nonlinearities —and most of the usual chaotic systems are described in that way—, there exist several methods of elimination, one of the most popular being the Gröbner bases method. After having obtained the input/output relation, we can resort to the following proposition:

*Proposition 1 ([23]):* The system (9) (or (10)) is structurally identifiable if, for almost every parameter vector $\theta = [\theta^{(1)}, \ldots, \theta^{(L)}]$, from the input/output relation, the parameters $\theta^{(i)}$ can be rearranged in a linear regression such that, for $i = 1, \ldots, L$,

$$\begin{aligned} &P_i(y_k, \ldots, y_{k+N}, m_k, \ldots, m_{k+N})\theta^{(i)} \\ &-Q_i(y_k, \ldots, y_{k+N}, m_k, \ldots, m_{k+N}) = 0, \quad i = 1, \ldots, L, \end{aligned} \tag{16}$$

where $P_i$ and $Q_i$ are polynomials depending on $y_k$, $m_k$ and on their iterates.

$N$ depends on the so-called observability index of the system and on $L$. Since the notion of observability index is outside of the main scope of the present paper, we refer the reader to [34], P.157 for further information.

Based upon the Proposition 1, it is a simple matter to infer that, if the parameter vector (or equivalently the system) is structurally identifiable, $\theta^{(i)}$ can be written as

$$\theta^{(i)} = \frac{Q_i(y_k, \ldots, y_{k+N}, m_k, \ldots, m_{k+N})}{P_i(y_k, \ldots, y_{k+N}, m_k, \ldots, m_{k+N})}. \tag{17}$$

Assessing the security through the concept of parametric identifiability leads to the following paradox: the most secure situation occurs when there exists a unique value of the parameter $\theta$ which induces a prescribed behavior at the output, what amounts to $\theta$ being structurally identifiable. On the other hand, it is computationally easy for polynomial systems to recover an identifiable parameter from a sufficient amount of input/output pairs $(m_k, y_k)$ (provided that a known plaintext attack, meaning that $m_k$ is known, can be performed). Indeed there exist powerful tools for eliminating the internal state $x_k$ from (9) or (10) and solving the resulting equations in $\theta$ to obtain (17). Retrieving the parameter in this way constitutes an *algebraic attack*.

A fundamental conclusion is that message-embedded chaotic cryptosystems involving only polynomial nonlinearities are weak against algebraic attacks and need extensive elaboration. Alternate solutions are presented in the subsequent sections.

### D. Two-channel transmission

For a two-channel transmission (Fig. 6), a first channel is used to convey the output $y_k$ of an autonomous chaotic system with dynamic $f_\theta$ and output function $h_\theta$. Besides, a function $\nu_e$, depending on a time-varying quantity, say, the state vector $x_k$ of the chaotic system, encrypts the information $m_k$ and delivers $u_k = \nu_e(x_k, m_k)$. Then, the signal $u_k$ is transmitted via a second channel. The set of equations governing the transmitter is

$$\begin{cases} x_{k+1} = f_\theta(x_k) \\ y_k = h_\theta(x_k) \\ u_k = \nu_e(x_k, m_k) \end{cases} . \tag{18}$$

At the receiver end, since the chaotic signal $y_k$ is information-free, a perfect synchronization fulfilling (1) or (2) can be achieved. As a consequence, the information $m_k$ can be correctly recovered by:

$$\begin{cases} \hat{x}_{k+1} = \tilde{f}_\theta(\hat{x}_k, y_k) \\ \hat{y}_k = \tilde{h}_\theta(\hat{x}_k) \\ \hat{m}_k = \nu_d(\hat{x}_k, u_k) \end{cases} , \tag{19}$$

which are the equations governing the receiver and may implement an observer. The decrypting function $\nu_d$ is defined by

$$\hat{m}_k = \nu_d(\hat{x}_k, u_k) = m_k \text{ when } \hat{x}_k = x_k. \tag{20}$$

This technique has been proposed, for example, in [33][45]. The advantage lies in that, unlike modulation-based approaches, $m_k$ is allowed to switch every discrete times $k$ without inducing synchronization transients. Only the first values of $\hat{m}_k$, before the synchronization might be achieved, will be wrong. On the other hand, a transmission involving two channels may be unsatisfactory for throughput purposes.

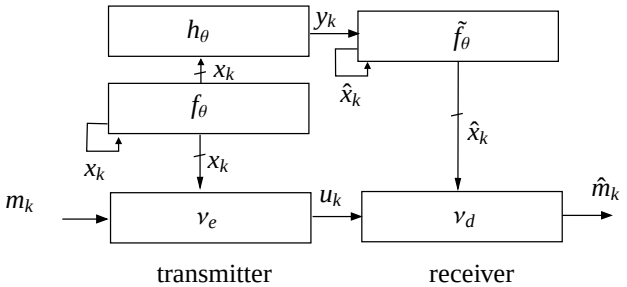


Fig. 6. Two-channel transmission

### E. Hybrid Message-embedding

The hybrid message-embedded technique (Fig. 7) was proposed in [44] and partially cryptanalyzed in [37] wherein the term "hybrid" was first introduced. It uses, at the transmitter side, the three same units as the ones involved in the two-channel transmission. On the other hand, $u_k = \nu_e(x_k, m_k)$ is not directly conveyed through the channel but is reinjected into the chaotic dynamics $f_\theta$ (and possibly into $h_\theta$). Only the output $y_k$, that implicitly or explicitly depends on $u_k$ and so on $m_k$, is transmitted. Basically, the hybrid message-embedded technique encompasses the simple message-embedded one. In fact, it can be deduced in a straightforward way by replacing $m_k$ by $\nu_e(x_k, m_k)$. Conversely, when $\nu_e$ is such that $\nu_e(x_k, m_k) = m_k$ for all $x_k$, the hybrid message-embedded cryptosystem comes down to the message-embedded one. We distinguish two different setups. The first one corresponds to transmitter systems having relative degree $r = 0$ with respect to $u_k$:

$$\begin{cases} x_{k+1} = f_\theta(x_k, u_k) \\ y_k = h_\theta(x_k, u_k) \\ u_k = \nu_e(x_k, m_k) \end{cases} , \tag{21}$$

while the second class corresponds to systems having relative degree $r > 0$ with respect to $u_k$:

$$\begin{cases} x_{k+1} = f_\theta(x_k, u_k) \\ y_k = h'_\theta(x_k) \\ u_k = \nu_e(x_k, m_k) \end{cases} . \tag{22}$$

*Remark 2:* Sometimes, owing to the similarity with the two-channel transmission, the quantity $u_k$ is also called the ciphertext. As a matter of fact, this is a misleading terminology since for the hybrid message-embedding, $u_k$ turns into a mere internal variable. The actual ciphertext is related to $y_k$ which is conveyed through the channel and is publicly accessible. $u_k$ may almost be called the pre-ciphertext.

Similarly to what happened with the message-embedding technique, after iterating $r$ times the state vector in (22), the output $y_{k+r}$ reads

$$y_{k+r} = h'_\theta(f^r_\theta(x_k, u_k)), \tag{23}$$

where

$$\begin{aligned} f^i_\theta(x_k, u_k) &= x_k \text{ when } i = 0 \\ &= f_\theta(f^{i-1}_\theta(x_k, u_k), u_{k+i-1}) \quad \forall i \geq 1 \end{aligned} .$$

In (23), $u_k$ appears explicitly according to the Definition 1 while replacing $m_k$ by $u_k$.

The *receiver* system must be designed in such a way that both $u_k$ and $x_k$ can be recovered, given the only available data $y_k$ and its subsequent iterates. Once $u_k$ is recovered, the plaintext $m_k$ is correctly extracted by applying the decryption function $\nu_d$, provided that $\hat{x}_k$ is exactly synchronized with $x_k$. Similarly to the message-embedding technique, the synchronization and recovering of $u_k$ can resort to an inverse system or to an unknown input observer of the form

$$\begin{cases} \hat{x}_{k+1} = \tilde{f}_\theta(\hat{x}_k, y_k, \ldots, y_{k+r}) \\ \hat{u}_k = g_\theta(\hat{x}_k, y_{k+r}) \\ \hat{m}_k = \nu_d(\hat{x}_k, \hat{u}_k) \end{cases} \tag{24}$$

with $g$ such that

$$\hat{u}_k = g_\theta(\hat{x}_k, y_{k+r}) = u_k \text{ when } \hat{x}_k = x_k \tag{25}$$

and with $\nu_d$ such that

$$\hat{m}_k = \nu_d(\hat{x}_k, \hat{u}_k) = m_k \text{ when } \hat{x}_k = x_k \text{ and } \hat{u}_k = u_k. \quad (26)$$

Unlike the two-channel transmission or the parameter modulation, the hybrid message-embedded technique, similarly to the mere message-embedding, offers the advantages that only a single channel is needed and, moreover, that the synchronization can be guaranteed without restriction on the rate of variation of $m_k$. Additionally, the scheme allows to introduce a highly nonlinear function $\nu_e$, possibly non-polynomial, which can make the state generator significantly more resistant to algebraic attacks, as described in Subsection II-C.2.

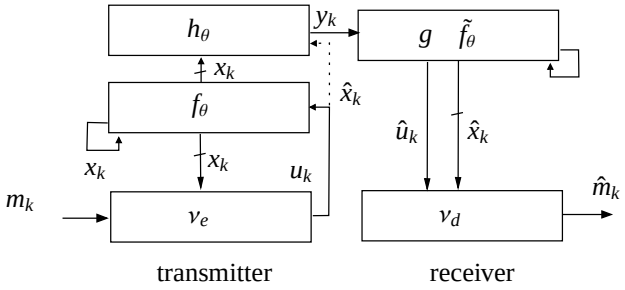


Fig. 7. Hybrid message-embedding

## III. Comparative study

This section (the second part of the paper) deals with the connection between chaotic cryptosystems and symmetric conventional encryption. For details on conventional cryptography, see the book of Menezes [28], that has become a standard reference. However, we believe that, with the background reviewed hereafter, this section is sufficiently self-contained so as to be understood without further reading.

### A. Conventional cryptography

Among a wide variety of cryptographic techniques, two major classes can be typically distinguished: *public-key* ciphers and *symmetric-key* ciphers (also called private-key ciphers).

Public-key ciphers are largely based upon *trapdoor one-way functions*. These functions are defined from a set $X$ to a set $Y$ in such a way that the computation of the image of $x \in X$ under $f$, denoted $f(x)$, is "easy", whereas the search for $x$ from only the knowledge of the image $y = f(x) \in Y$, is computationally infeasible unless some extra information is provided. This extra information, making feasible to compute $x$ from $f(x)$, is referred to as the trapdoor information. Usual trapdoor one-way functions are based on computationally very demanding mathematical problems, for instance, prime factorization. Similarly, a public-key cipher has the property that retrieving the key $k^d$ (associated to the decryption function $d$) from only the knowledge of the key $k^e$ (associated to the encryption function $e$) is exceedingly time-consuming with the current mathematical algorithms and, most importantly, with the available computation power. Consequently, only the private key $k^d$ must be kept secret, while the public key $k^e$ may be known to any (authorized or unauthorized) user of the communication network. $k^d$ plays the role of the trapdoor information. One of the best well-known public-key ciphers is the RSA algorithm [28].

In contrast to public-key ciphers, symmetric-key ciphers are characterized by an encryption scheme $(e, d)$ such that the determination of the key $k^d$ can be easily done from the knowledge of $k^e$. Hence, not only $k^d$ must be kept secret but the key $k^e$ as well. It is customary that both keys are identical, that is, $k^d = k^e$. There are two classes of symmetric-key encryption schemes which are commonly distinguished: block ciphers and stream ciphers.

A block cipher is an encryption scheme that breaks up the plaintext messages into strings (called blocks) of a fixed length over an alphabet and encrypts one block at a time. Block ciphers usually involve *substitution* ciphers, *transposition* ciphers or, more generally, *product* ciphers, which are compositions of the former ones.

Stream ciphers are described in more details in the next subsection.

*1) Stream ciphers:* In the case of stream ciphers, the *plaintext* is broken up into blocks of the same length, called symbols and denoted by $m_k$. A major distinction with respect to the block ciphers lies in that the encryption function $e$ can change for each symbol because it depends on a time-varying key $K_k$. The sequence $\{K_k\}$ is called the *keystream*. If $m_k$ is the $k^{th}$ symbol of the plaintext at time $k$, each element $c_k$ of the ciphertext obeys at time $k$

$$c_k = e(K_k, m_k).$$

This being the case, stream ciphers require a keystream generator. It is customary that the plaintext $m_k$ and the ciphertext $c_k$ are binary words, the most widely adopted function $e$ performing a mere bitwise XOR operation. If the keystream $\{K_k\}$ would be truly random and never used again, the encryption scheme would be called *one-time pad* —the only cipher provably secure so far. However, in order to decrypt the ciphertext, the recipient party would have to know the random keystream and, thus, would require again a secure transmission of the key; this is the so-called key-exchange problem which can be solved in different ways, notably via public-key cryptography. However, for the *one-time pad* cipher, the key should be as long as the plaintext which drastically increases the difficulty of the key distribution. As an alternative to such an ideal encryption scheme, one can resort to pseudo-random generators. Indeed, for such generators, the keystream is produced by a deterministic function while its statistical properties look random. Generally, keystreams are generated iteratively by feedback shift registers since they produce pseudo-random sequences in a very efficient way [20]. There are two classes of stream ciphers, the difference lying in the way the keystream is generated: the *synchronous* stream ciphers (SSC) and the *self-synchronous* stream ciphers (SSSC).

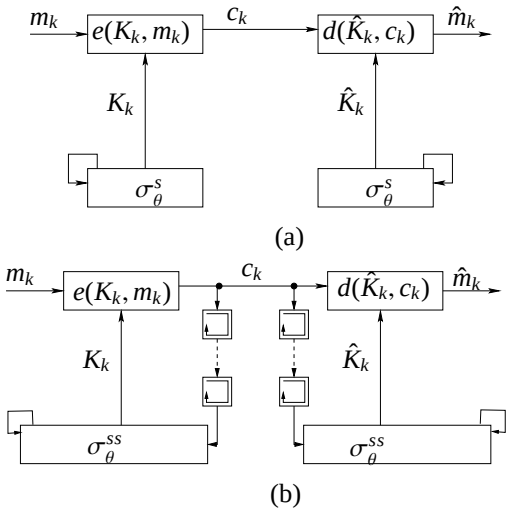


Fig. 8. Stream cipher: (a) synchronous, (b) self-synchronous

A block diagram of SSC is given in Fig.8(a). The equations of the transmitter are:

$$\begin{cases} K_{k+1} = \sigma^s_\theta(K_k) \\ c_k = e(K_k, m_k) \end{cases} . \tag{27}$$

The key $K_k$ is generated by a function $\sigma^s_\theta$ parameterized by $\theta$, the parameter $\theta$ acting as the secret *static* (or master) key. The ciphertext $c_k$ is available at the transmitter output and conveyed through the channel.

A block diagram of SSSC is given in Fig.8(b). Actually, this is a conceptual model called canonical representation, that can correspond to numerous different architectures. The SSSC admits, at the transmitter side, the recursions

$$\begin{cases} K_{k+1} = \sigma^{ss}_\theta(c_{k-l}, \ldots, c_{k-l'}) \\ c_k = e(K_k, m_k) \end{cases} , \tag{28}$$

where $\sigma^{ss}_\theta$ is also a function parameterized by $\theta$ that generates the keystream $\{K_k\}$. Unlike SSC, $K_k$ does not depend now on an internal dynamic but only on a fixed number of past values of $c_k$; The quantity $b = |l - l' + 1|$ is called the *memory*; most often one has $l = 0$. As previously, $c_k$ is generated by the encryption function $e$ depending on a time-varying key $K_k$.

For both SSC and SSSC, the reconstruction of the plaintext requires the synchronization of the two sequences $\{K_k\}$ and $\{\hat{K}_k\}$ produced at the transmitter and the receiver sides. The inherent determinism allows their synchronization as detailed next.

In the SSC case, the decryption is specified at the receiver part by

$$\begin{cases} \hat{K}_{k+1} = \sigma^s_\theta(\hat{K}_k) \\ \hat{m}_k = d(\hat{K}_k, c_k) \end{cases} \tag{29}$$

and, in the SSSC case, by

$$\begin{cases} \hat{K}_{k+1} = \sigma^{ss}_\theta(c_{k-l}, \ldots, c_{k-l'}) \\ \hat{m}_k = d(\hat{K}_k, c_k) \end{cases} . \tag{30}$$

In both cases, the decryption function $d$ is such that

$$\hat{m}_k = d(\hat{K}_k, c_k) = m_k \text{ when } \hat{K}_k = K_k. \tag{31}$$

For the SSC, the keystreams $\{K_k\}$ and $\{\hat{K}_k\}$ result from autonomous recurrences. It turns out that the unique way of achieving the synchronization is to initialize the key of the generators $\sigma^s_\theta$ at both sides at the same value ($\hat{K}_0 = K_0$). Therefore, $K_0$ acts as the secret static key, that is, $\theta = K_0$.

As for the SSSC, $\theta$ is the parameter vector of the function $\sigma^{ss}$. If the parameters are identical at both sides, the respective keystreams synchronize automatically because $\sigma^{ss}_\theta$ operate, at both sides, on the same quantities, namely the past values of $c_k$. The ability to self-synchronizing constitutes one of the main advantages of such cryptosystems. Indeed, they are resistant against bit slips on the transmission channel without any additional synchronization flags or interactive protocols for recovering lost synchronization.

### *B. A connection*

A major and obvious difference between chaotic ciphers and stream ciphers consists in that a chaotic generator is assumed to produce an aperiodic sequence $\{x_k\}$ ranging in a dense set, while the pseudo-random generators used in stream ciphers produce discrete sequences. Yet, observe that when chaotic generators are implemented in machines with finite accuracy (say, a computer), the sequences $\{x_k\}$ are not really chaotic. Indeed, since the set on which the $x_k$'s take values has finite cardinality, such sequences will obviously get trapped in a loop, called *cycle*, of finite period. We can expect this period to be not too short and the degree of 'randomness' of the sequence to be high (as measured e.g. by standard statistical tests), but guaranteeing the said properties requires some caution [20]. Important contributions to this issue and a definition of *discrete chaos* can be found in [22]. Henceforth we focus rather on the structure of the proposed setups for the comparative study, regardless of the dynamic involved. Let us notice that we however still use the terminology chaotic cryptosystem to distinguish them from the conventional ones. The main results of this survey are contained in Proposition 1 and Proposition 2 of this section. They state that message-embedding and hybrid message-embedding are strictly equivalent to conventional self-synchronizing stream ciphers under flatness conditions.

*1) Additive masking:* A natural connection can be made between additive masking and SSC. In fact, the transmitter of the respective schemes has exactly the same structure. The sequences $\{x_k\}$ for chaotic cryptosystems (*resp.* $\{K_k\}$ for SSC) are independent from the plaintext $m_k$ and the ciphertext $y_k$ (*resp.* $c_k$). For a SSC, the same initialization is required at both ends to assure synchronization. For additive masking and assuming that the generator is really chaotic, synchronization would be inevitably lost within a very short time window due to sensitivity to initial conditions. To handle such a problem, a controlled synchronization at the receiver part usually based on observers, is often suggested as mentioned in Section II-A. Nevertheless, as previously

pointed out, the added information to be masked acts as a disturbance and prevents the control from guaranteeing an exact synchronization. This renders the additive masking not very appealing compared with a conventional SSC.

*2) Message-embedding:* The results stated in this section are based on the notion of *flatness* (see [25] for an introductory theory)

*Definition 5:* (Flatness) A system with dynamic $f$, parametrized by $\theta$, of relative degree $r$, input $e_k$ and state vector $z_k$ of dimension $n$ is said to be *flat* if there exists a set of independent variables $y_k$, referred to as flat outputs, such that all 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, for Single Input Single Output systems, there exist two functions $\mathcal{F}_\theta$ and $\mathcal{G}_\theta$ which obey

$$\left\{ \begin{array}{rcl} z_k & = & \mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(r)}, \ldots, y_{k+k'_{\mathcal{F}_\theta}(r)}) \\ e_k & = & \mathcal{G}_\theta(y_{k+k_{\mathcal{G}_\theta}(r)}, \ldots, y_{k+k'_{\mathcal{G}_\theta}(r)}) \end{array} \right. . \tag{32}$$

where $k_{\mathcal{F}_\theta}(r)$, $k'_{\mathcal{F}_\theta}(r)$, $k_{\mathcal{G}_\theta}(r)$ and $k'_{\mathcal{G}_\theta}(r)$ are $\mathbb{Z}$-valued integers depending on the relative degree $r$ of the system.

Now we can state the following proposition.

*Proposition 2:* The message-embedding cryptosystem (9) (or (10)) is equivalent to a conventional self-synchronizing stream cipher if the nonlinear dynamic $f$ with output $y_k$ and input $m_k$ is flat.

We provide below a constructive proof.

*Proof:* According to the Definition 5, flatness of (9), with relative degree $r = 0$, means that there exist two functions $\mathcal{F}_\theta$ and $\mathcal{G}_\theta$ and integers $k_{\mathcal{F}_\theta}(0)$, $k'_{\mathcal{F}_\theta}(0)$, $k_{\mathcal{G}_\theta}(0)$ and $k'_{\mathcal{G}_\theta}(0)$ such that

$$\begin{array}{rcl} x_k & = & \mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(0)}, \ldots, y_{k+k'_{\mathcal{F}_\theta}(0)}) \\ m_k & = & \mathcal{G}_\theta(y_{k+k_{\mathcal{G}_\theta}(0)}, \ldots, y_{k+k'_{\mathcal{G}_\theta}(0)}) \end{array} . \tag{33}$$

After iterating once forward the first equation of (33), it turns out that (9) is strictly equivalent to

$$\left\{ \begin{array}{l} x_{k+1} = \mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(0)+1}, \ldots, y_{k+k'_{\mathcal{F}_\theta}(0)+1}) \\ y_k = h_\theta(x_k, m_k) \end{array} \right. . \tag{34}$$

Identifying (34) with (28) leads to the following result:

*i)* The system (9) is strictly equivalent to the transmitter part of a self-synchronizing stream cipher of the form (28) with key generator $\sigma_\theta^{ss} = \mathcal{F}_\theta$, running key $K_k = x_k$, ciphertext $c_k = y_k$, encrypting function $e = h$, secret static key $\theta$ and memory $b = |k_{\mathcal{F}_\theta}(0) - k'_{\mathcal{F}_\theta}(0) + 1|$.

*Remark 3:* An alternate but equivalent way of obtaining $x_{k+1}$ in (34) consists in substituting $x_k$ and $m_k$ of (33) into the dynamical equation of (9) yielding

$$x_{k+1} = f_\theta(\mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(0)}, \ldots, y_{k+k'_{\mathcal{F}_\theta}(0)}), \mathcal{G}_\theta(y_{k+k_{\mathcal{G}_\theta}(0)}, \ldots, y_{k+k'_{\mathcal{G}_\theta}(0)})). \tag{35}$$

Besides, according to Definition 5, flatness of (10), with relative degree $r > 0$, means that there exist two functions $\mathcal{F}_\theta$ and $\mathcal{G}_\theta$ and integers $k_{\mathcal{F}_\theta}(r)$, $k'_{\mathcal{F}_\theta}(r)$, $k_{\mathcal{G}_\theta}(r)$ and $k'_{\mathcal{G}_\theta}(r)$ such that

$$\left\{ \begin{array}{rcl} x_k & = & \mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(r)}, \ldots, y_{k+k'_{\mathcal{F}_\theta}(r)}) \\ m_k & = & \mathcal{G}_\theta(y_{k+k_{\mathcal{G}_\theta}(r)}, \ldots, y_{k+k'_{\mathcal{G}_\theta}(r)}) \end{array} \right. . \tag{36}$$

After iterating once forward the first equation of (36) and taking into account (11), it turns out that (10) is strictly equivalent to

$$\left\{ \begin{array}{l} x_{k+1} = \mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(r)+1}, \ldots, y_{k+k'_{\mathcal{F}_\theta}(r)+1}) \\ y_{k+r} = h'_\theta(f_\theta^r(x_k, m_k)) \end{array} \right. . \tag{37}$$

Letting $l_{h',f^r}(x_k, m_k) = h'_\theta(f_\theta^r(x_k, m_k))$ since $y_{k+r}$ depends explicitly on $x_k$ and $m_k$, identification of (37) with (28) leads then to the following result:

*ii)* The system (10) is strictly equivalent to the transmitter part of a self-synchronizing stream cipher of the form (28) with key generator $\sigma_\theta^{ss} = \mathcal{F}_\theta$, running key $K_k = x_k$, ciphertext $c_k = y_{k+r}$, encrypting function $e = l_{h',f^r}$, secret static key $\theta$ and memory $b = |k_{\mathcal{F}_\theta}(r) - k'_{\mathcal{F}_\theta}(r) + 1|$.

This completes the proof. ■

*Remark 4:* It is worthwhile noticing that the set of equations (33) (*resp.* (36)) could be used at the receiver part to obtain both $x_k$ and $m_k$ without resorting to a state reconstruction through an inverse system or an Unknown Input Observer like (24). Even more is true: since one has

$$m_k = \mathcal{G}_\theta(y_{k+k_{\mathcal{G}_\theta}(r)}, \ldots, y_{k+k'_{\mathcal{G}_\theta}(r)}), \tag{38}$$

with $r = 0$ (*resp.* $r > 0$), the message $m_k$ can be retrieved in finite time and the knowledge of $x_k$ is no longer useful. However, given a system, the difficulty lies in finding out the quantities $k_{\mathcal{F}_\theta}(r)$, $k'_{\mathcal{F}_\theta}(r)$, $k_{\mathcal{G}_\theta}(r)$ and $k'_{\mathcal{G}_\theta}(r)$ and writing down explicitly the functions $\mathcal{F}_\theta$ and $\mathcal{G}_\theta$. It can be shown (see [7] for the linear case) that resorting to a state space approach actually allows to achieve this computation in an implicit and recursive way. Indeed, for flat systems only a finite number of iterations of (24) is needed to achieve $\hat{x}_k = x_k$. It turns out that the resulting state vector $\hat{x}_k = x_k$ only depends on past values of $y_k$, which provides $\mathcal{F}_\theta$ in (32). Then, substituting $\hat{x}_k = x_k$ into (25) provides $\mathcal{G}_\theta$ in (32).

*3) Hybrid Message-embedding:* We refer again to Definition 5 for the following proposition.

*Proposition 3:* The hybrid message-embedding cryptosystem (21) (*or* (22)) is equivalent to a conventional self-synchronizing stream cipher if the nonlinear dynamic $f$ with output $y_k$ and input $u_k$ is flat.

*Proof:* Flatness of (21), with relative degree $r = 0$, means that there exist two functions $\mathcal{F}_\theta$ and $\mathcal{G}_\theta$ and integers $k_{\mathcal{F}_\theta}(0)$, $k'_{\mathcal{F}_\theta}(0)$, $k_{\mathcal{G}_\theta}(0)$ and $k'_{\mathcal{G}_\theta}(0)$ such that

$$\begin{array}{rcl} x_k & = & \mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(0)}, \ldots, y_{k+k'_{\mathcal{F}_\theta}(0)}) \\ u_k & = & \mathcal{G}_\theta(y_{k+k_{\mathcal{G}_\theta}(0)}, \ldots, y_{k+k'_{\mathcal{G}_\theta}(0)}) \end{array} . \tag{39}$$

When iterating once forward the first equation of (39), it turns out that (21) is strictly equivalent to

$$\left\{\begin{array}{lcl} x_{k+1} & = & \mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(0)+1},\ldots,y_{k+k'_{\mathcal{F}_\theta}(0)+1}) \\ y_k & = & h_\theta(x_k,\nu_e(x_k,m_k)) \end{array}\right. . \quad (40)$$

Letting $l_{h,\nu_e}(x_k,m_k) = h_\theta(x_k,\nu_e(x_k,m_k))$ since $y_k$ depends explicitly on $x_k$ and $m_k$, identification of (40) with (28) leads then to the following result:

*iii)* The system (21) is strictly equivalent to the transmitter part of a self-synchronizing stream cipher of the form (28) with key generator $\sigma_\theta^{ss} = \mathcal{F}_\theta$, running key $K_k = x_k$, ciphertext $c_k = y_k$, encrypting function $e = l_{h,\nu_e}$, secret static key $\theta$ and memory $b = |k_{\mathcal{F}_\theta}(0) - k'_{\mathcal{F}_\theta}(0) + 1|$.

Besides, flatness of (22), with relative degree $r > 0$, means that there exist two functions $\mathcal{F}_\theta$ and $\mathcal{G}_\theta$ and integers $k_{\mathcal{F}_\theta}(r)$, $k'_{\mathcal{F}_\theta}(r)$, $k_{\mathcal{G}_\theta}(r)$ and $k'_{\mathcal{G}_\theta}(r)$ such that

$$\begin{array}{lcl} x_k & = & \mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(r)},\ldots,y_{k+k'_{\mathcal{F}_\theta}(r)}) \\ u_k & = & \mathcal{G}_\theta(y_{k+k_{\mathcal{G}_\theta}(r)},\ldots,y_{k+k'_{\mathcal{G}_\theta}(r)}) \end{array} . \quad (41)$$

When iterating once forward the first equation of (41) and taking into account (23), it turns out that (22) is strictly equivalent to:

$$\left\{\begin{array}{lcl} x_{k+1} & = & \mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(r)+1},\ldots,y_{k+k'_{\mathcal{F}_\theta}(r)+1}) \\ y_{k+r} & = & h'_\theta(f^r_\theta(x_k,\nu_e(x_k,m_k))) \end{array}\right. . \quad (42)$$

Letting $l_{h',f^r,\nu_e}(x_k,m_k) = h'_\theta(f^r_\theta(x_k,\nu_e(x_k,m_k)))$ since $y_{k+r}$ depends explicitly on $x_k$ and $m_k$, identification of (42) with (28) leads then to the following result:

*iv)* The system (22) is strictly equivalent to the transmitter part of a self-synchronizing stream cipher of the form (28) with key generator $\sigma_\theta^{ss} = \mathcal{F}_\theta$, running key $K_k = x_k$, ciphertext $c_k = y_{k+r}$, encrypting function $e = l_{h',f^r,\nu_e}$, secret static key $\theta$ and memory $b = |k_{\mathcal{F}_\theta}(r) - k'_{\mathcal{F}_\theta}(r) + 1|$.

This completes the proof. ■

*Remark 5:* It is worthwhile noticing that the set of equations (39) (*resp.* (41)) could be used at the receiver part to obtain both $x_k$ and $u_k$ without resorting to a state reconstruction through an inverse system or an Unknown Input Observer like (12). Moreover, substituting $x_k$ and $u_k$ of (39) (*resp.* (41)) into (20) yields

$$m_k = \nu_d(\mathcal{F}_\theta(y_{k+k_{\mathcal{F}_\theta}(r)},\ldots,y_{k+k'_{\mathcal{F}_\theta}(r)}),\mathcal{G}_\theta(y_{k+k_{\mathcal{G}_\theta}(r)},\ldots,y_{k+k'_{\mathcal{G}_\theta}(r)})), \quad (43)$$

with $r = 0$ (*resp.* $r > 0$), so that the message $m_k$ can be retrieved in finite time and the knowledge of $x_k$ is no longer useful. However, resorting to a state space approach is motivated in the same manner as in Remark 4 for the message-embedding.

The comparison between Eq. (43) and Eq. (38) highlights the way how, by introducing the function $\nu_e$, this scheme becomes more complex than the message-embedding.

*4) Example:* This example illustrates the aforementioned connection between the hybrid message-embedding and self-synchronizing cryptosystems. We consider a 3-dimensional linear congruential hybrid message-embedded cryptosystem like (22) with dynamic $f$ and output function $h'$ of the form:

$$\left\{\begin{array}{lcl} x_{k+1} & = & Ax_k + Bu_k \\ y_k & = & Cx_k \\ u_k & = & \nu_e(x_k,m_k) \end{array}\right. . \quad (44)$$

The entries of the matrices $A$, $B$ and $C$ are integers ranging between 0 and 255, the modulo being $m = 256$. All along this section, the operations are performed modulo $m$. Numerically, the matrices read

$$A = \left[\begin{array}{ccc} 38 & 1 & 0 \\ 7 & 0 & 1 \\ 4 & 0 & 0 \end{array}\right], \quad B = \left[\begin{array}{c} 1 \\ 0 \\ 0 \end{array}\right],$$

$$C = \left[\begin{array}{ccc} 1 & 0 & 0 \end{array}\right].$$

It is recalled that, for linear systems written in a state space form, the relative degree corresponds to the smallest integer $r$ such that $CA^{r-1}B$ is different from 0 ([18]). Here, since $CB = 1$, the relative degree of the system is 1. The supposed secret static key is the vector $\theta = [38\ 7\ 4]$ which actually corresponds to the first column of $A$ written in a companion form. The function $\nu_e$ is chosen to be a bitwise XOR (denoted $\oplus$) between the components of $x_k$ denoted $x_k^{(i)}$ and the plaintext $m_k$:

$$u_k = x_k^{(1)} \oplus x_k^{(2)} \oplus x_k^{(3)} \oplus m_k.$$

where $x_k^{(i)}$ and $m_k$ are meant here to be the corresponding 8-bit representation. It turns out that after iterating three times the inverse system of (44) (the structure is not provided here but see for example [7] for details), as mentioned in the Remark 4, we obtain the equations in the form (41) ) with $\mathcal{F}_\theta$ obeying

$$\left\{\begin{array}{lcl} x_k^{(1)} & = & y_k \\ x_k^{(2)} & = & 7y_{k-1} + 4y_{k-2} \\ x_k^{(3)} & = & 4y_{k-1} \end{array}\right. \quad (45)$$

and the function $\mathcal{G}_\theta$ obeying

$$u_k = y_{k+1} - 38y_k - 7y_{k-1} - 4y_{k-2} . \quad (46)$$

Equations (45) and (46) clearly corroborate that the system is flat. Besides, they provide the actual values $k_{\mathcal{F}_\theta}(1) = 0$, $k'_{\mathcal{F}_\theta}(1) = -2$, $k_{\mathcal{G}_\theta}(1) = 1$ and $k'_{\mathcal{G}_\theta}(1) = -2$ The relative degree $r$ of the system being 1, we must compute $y_{k+1}$.

$$\begin{array}{lcl} y_{k+1} & = & CAx_k + CB\nu_e(x_k,m_k) \\ & = & 38x_k^{(1)} + x_k^{(2)} + x_k^{(1)} \oplus x_k^{(2)} \oplus x_k^{(3)} \oplus m_k \\ & = & l_{h',f^1,\nu_e}(x_k,m_k) \end{array} \quad (47)$$

Iteration of (45) once forward and consideration of (47) allow us to claim the result *iv)*:

The system (44) is strictly equivalent to the transmitter part of a self-synchronizing stream cipher of the form (28) with key generator $\sigma_\theta^{ss} = \mathcal{F}_\theta$ corresponding to Eq. (45), running key $K_k = x_k$, ciphertext $c_k = y_{k+1}$, encrypting function $e = l_{h',f^1,\nu_e}$ corresponding to Eq. (47), secret static key $\theta = [38\ 7\ 4]$ and memory $b = 2 + 1 = 3$.

Retrieving $m_k$ requires the computation (43). Here the function $\nu_d$ is also an XOR between the components of $x_k$ and the (pre-)ciphertext $u_k = \nu_e(x_k, m_k)$, that is, $\nu_d(x_k, u_k) = u_k \oplus x_k^{(1)} \oplus x_k^{(2)} \oplus x_k^{(3)}$ where $x_k^{(i)}$ and $u_k$ are meant to be the corresponding 8-bit representation similarly to the function $\nu_e$. Indeed, $u_k \oplus x_k^{(1)} \oplus x_k^{(2)} \oplus x_k^{(3)} = m_k \oplus x_k^{(1)} \oplus x_k^{(2)} \oplus x_k^{(3)} \oplus x_k^{(1)} \oplus x_k^{(2)} \oplus x_k^{(3)} = m_k$. The system being flat, $x_k^{(i)}$ can be expressed in terms of delayed outputs as indicated by the function $\mathcal{F}_\theta$. Hence, one has

$$m_k = (y_{k+1} - 38y_k - 7y_{k-1} - 4y_{k-2}) \oplus y_k \oplus (7y_{k-1} + 4y_{k-2}) \oplus 4y_{k-1}.$$

Let us notice that if we simplify the hybrid message-embedded cryptosystem to a message-embedded one (Eq. 9)) , in other words, if $u_k$ is substituted by $m_k$ ($\nu_e$ and so $\nu_d$ are no longer used), then the computation of $m_k$ reduces to

$$m_k = y_{k+1} - 38y_k - 7y_{k-1} - 4y_{k-2}.$$

The foregoing input/output relation is clearly simpler and highlights the enhancement of the hybrid structure.

## IV. Conclusion

In this paper we have reviewed some basic architectures of chaotic and conventional stream ciphers, thereby establishing a formal parallelism. We may sum up the main conclusions in the following points.

- (Hybrid) message-embedding seems to be the most efficient chaotic cryptosystem in practice. For a non flat transmitter part, the decryption requires an inverse system or an observer achieving an asymptotical recovering of the plaintext. But if a finite time convergence is sought, then the transmitter part must implement a flat dynamics. In this case, the resulting cryptosystem is strictly equivalent to a conventional self-synchronizing stream cipher.

- As for cryptographical security, we conclude, based on the parallelism mentioned above, that digital (hybrid) message-embedding is able to provide the same security as any conventional self-synchronizing stream cipher. Needless to say, several stream ciphers (e.g., RC4) are currently being used in, say, internet and mobile communications. Since these ciphers are considered sufficiently secure for such purposes, the same consideration should be extended to message-embedded ciphers under a suitable choice of functions $f$, $h$ (or $h'$) and $\nu_e$, which may constitute a challenging task for further research.

**Acknowledgment** Jos´e Maria Amig´o was partially funded by the Spanish Ministery of Science and Education (Ref. MTM2005-04948). The work presented in this paper was also partially supported by a grant from the Agence Nationale pour la Recherche in France (Ref. ANR-05-JCJC-0112-01)

## References

[1] J.M. Amig´o, J. Szczepanski, and L. Kocarev. A chaos-based approach to the design of crytographically secure substitutions. *Phys. Lett. A*, 343:55–60, February 2005.

[2] F. Anstett, G. Millerioux, and G. Bloch. Global adaptive synchronization based upon polytopic observers. In *Proc. of IEEE International symposium on circuit and systems, ISCAS'04*, pages 728 – 731, Vancouver, Canada, May 2004.

[3] F. Ansttett, G. Millrioux, and G. Bloch. Message-embedded cryptosystems : cryptanalysis and identifiability. In *Proc. of the 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, Sevilla, Spain, December 12-15 2005.

[4] L. Boutat-Baddas, J. P. Barbot and¡ D. Boutat, and R. Tauleigne. Sliding mode observers and observability singularity in chaotic synchronization. *Mathematical Problems in Engineering*, (1):11–31, May 2004.

[5] M. Boutayeb, M. Darouach, and H. Rafaralahy. Generalized state-space observers for chaotic synchronization and secure communications. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 49(3):345–349, March 2002.

[6] Huijberts H. J. C., Nijmeijer H., and Willems R. System identification in communication with chaotic systems. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 47(6):800–808, 2000.

[7] J. Daafouz, M. Fliess, and G. Mill´erioux. Une approche intrinsque des observateurs linaires entres inconnues. In *Proc. of the Conférence Internationale Francophone d'Automatique*, Bordeaux, May 2006.

[8] F. Dachselt, K. Kelber, J. Vandewalle, and W. Schwarz. Chaotic versus classical stream ciphers – a comparative study. In *Proc. of Int. Symp. on Circuits and Systems ISCAS'98*, volume IV, pages 518–521, Monterey, June 1998.

[9] Inoue E. and Ushio T. Chaos communication using unknown input observers. *Electronics and communication in Japan part III: Fundamental Electronic Science*, 84(12):21–27, 2001.

[10] A. L. Fradkov and A. Y. Markov. Adaptive synchronization of chaotic systems based on speed-gradient method and passification. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 44(10):905–912, Oct. 1997.

[11] J. Fridrich. Symmetric ciphers based on two-dimensional chaotic maps. *International Journal of Bifurcation and Chaos*, 8(6):1259 – 1284, June 1998.

[12] Kolumban G., Kennedy M. P., and Chua L. O. The role of synchronization in digital communications using chaos - part ıı: Chaotic modulation and chaotic synchronization. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 45:1129–1140, November 1998.

[13] G. Grassi and S. Mascolo. Nonlinear observer design to synchronize hyperchaotic systems via a scalar signal. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 44(10):1011–1014, Oct. 1997.

[14] Dedieu H. and Ogorzalek M. Identification of chaotic systems based on adaptive synchronization. In *Proc. ECCTD'97*, pages 290–295, Budapest, Sept. 1997.

[15] Dedieu H., Kennedy M. P., and Hasler M. Chaos shift keying: modulation and demodulation of a chaotic carrier using self-synchronizing chua's circuits. *IEEE Trans. Circuits. Syst. II: Anal. Digit. Sign. Process*, 40:634–642, 1993.

[16] Nijmeijer H. and Mareels I. M. Y. An observer looks at synchronization. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 44:882–890, October 1997.

[17] M. Hasler. Synchronization of chaotic systems and transmission of information. *International Journal of Bifurcation and Chaos*, 8(4), April 1998.

[18] A. Isidori. *Nonlinear control systems*. Communications and control engineering series. Springer, 1995.

[19] Lian K-Y. and Liu P. Synchronization with message embedded for generalized lorenz chaotic circuits and its error analysis. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 47(9):1418–1424, 2000.

[20] D. E. Knuth. *The Art of Computer Programming, Vol. 2*. Addison-Wesley, Reading, MA, 1998.

[21] L. Kocarev. Chaos-based cryptography :a brief overview. *IEEE Circuits and Systems Magazine*, 1(3):6–21, 2001.

[22] L. Kocarev, J. Szczepanski, J. M Amigo, and I. Tomosvski. Discrete chaos: part i. *IEEE Trans. on Circuits and Systems I*, 2006. in press.

[23] L. Ljung and T. Glad. On global identifiability for arbitarry model parametrizations. *Automatica*, 30(2):265–276, 1994.

[24] Cuomo K. M., Oppenheim A. V., and Strogatz S. H. Synchronization of lorenz-based chaotic circuits with applications to communications. *IEEE Trans. Circuits. Syst. II: Anal. Digit. Sign. Process*, 40(10):626–633, 1993.

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

[26] Itoh M., Wu C. W., and Chua L. O. Communications systems via chaotic signals from a reconstruction viewpoint. *International Journal of Bifurcation and Chaos*, 7(2):275–286, 1997.

[27] J.L. Massey. *Contemporary cryptology: an introduction*. G.J. Simmons, New York, ieee press edition, 1992.

[28] A. J. Menezes, P. C. Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, October 1996.

[29] G. Millerioux. Chaotic synchronization conditions based on control theory for systems described by discrete piecewise linear maps. *International Journal of Bifurcation and Chaos*, 7(7):1635–1649, 1997.

[30] G. Millerioux and J. Daafouz. An observer-based approach for input independent global chaos synchronization of discrete-time switched systems. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, pages 1270–1279, October 2003.

[31] G. Millerioux and J. Daafouz. Unknown input observers for message-embedded chaos synchronization of discrete-time systems. *International Journal of Bifurcation and Chaos*, 14(4):1357–1368, April 2004.

[32] G. Mill´erioux and J. Daafouz. *Chaos in Automatic Control*, chapter Polytopic observers for synchronization of chaotic maps, pages 323–344. Control Engineering Series. CRC, 2005.

[33] G. Millerioux and C. Mira. Coding scheme based on chaos synchronization from noninvertible maps. *International Journal of Bifurcation and Chaos*, 8(10):2019–2029, 1998.

[34] H. Nijmeijer and A. J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer, 1990.

[35] M. J. Ogorzalek. Taming chaos - part ı: synchronization. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 40(10):693–699, 1993.

[36] Palaniyandi P. and Lakshmanan M. Secure digital signal transmission by multistep parameter modumation and alternative driving of transmitter variables. *International Journal of Bifurcation and Chaos*, 11(7):2031–2036, 2001.

[37] A. T. Parker and K. M. Short. Reconstructing the keystream from a chaotic encryption scheme. *IEEE Trans. on Circ. and Syst.*, 48(5):624–630, May 2001.

[38] R. Schmitz. Use of chaotic dynamical systems in cryptography. *Journal of the Franklin Institute*, 338:429–441, 2001.

[39] J. Szczepanski, J.M. Amig´o, T. Michalek, and L. Kocarev. Crytographically secure substitutions based on the approximation of mixing maps. *IEEE Trans. Circuits and Systems I : Regular Papers*, 52(2):443–453, February 2005.

[40] Feldmann U., Hasler M., and Schwarz W. Communication by chaotic signals :the inverse system approach. *Int. J. of Circuit Theory Appl.*, 24:551–579, 1996.

[41] Parlitz U., Chua L. O., Kocarev L., Halle K. S., and Shang A. Transmission of digital signals by chaotic synchronization. *International Journal of Bifurcation and Chaos*, 3(2):973–977, 1993.

[42] Wu C. W. and Chua L. O. A simple way to synchronize chaotic systems with applications to secure communications systems. *International Journal of Bifurcation and Chaos*, 3(6):1619–1627, 1993.

[43] T. Yang. A survey of chaotic secure communication systems. *Int. J. of Computational Cognition*, 2004. (available at `http://www.YangSky.com/yangijcc.htm`).

[44] T. Yang, C. W. Wu, and L. O. Chua. Cryptography based on chaotic systems. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 44(5):469–472, May 1997.

[45] Jiang Z-P. A note on chaotic secure communication systems. *IEEE Trans. Circuits. Syst. I: Fundamental Theo. Appl*, 49(1):92–96, January 2002.