A connection between chaotic and conventional cryptography
Gilles Millérioux, Jose Maria Amigo, Jamal Daafouz

To cite this version:
Gilles Millérioux, Jose Maria Amigo, Jamal Daafouz. A connection between chaotic and conventional cryptography. IEEE Transactions on Circuits and Systems Part 1 Fundamental Theory and Applications, Institute of Electrical and Electronics Engineers (IEEE), 2008, 55 (6), pp.1695-1703. 10.1109/TCSI.2008.916555 . hal-00134924

HAL Id: hal-00134924
https://hal.archives-ouvertes.fr/hal-00134924
Submitted on 5 Mar 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 cryptography.

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.

A general encryption mechanism or scheme, also called cryptosystem or cipher, is illustrated in Fig. 1. We are given an alphabet , that is, a finite set of basic elements named symbols. On the transmitter part, a plaintext (also called information or message) is encrypted according to an encryption function which depends on the key . The resulting ciphertext is decrypted according to a decryption function which depends on the key . The function (resp. ) must

Fig. 1. General encryption mechanism

key source
transmitter
eavesdropper
m
k^e
\( e(k^e, m) \)
c
\( d(k^d, c) \)
\( m \)
\( k^d \)
key source
receiver
be a bijection from $\mathcal{M}$ to $\mathcal{C}$ (resp. $\mathcal{C}$ 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 $d(k^e, c) = m$ where $c = e(k^e, m)$. Let us point out that the design of a cryptographic scheme must take into account that the sets $\mathcal{M}, \mathcal{C}, \mathcal{K}$ 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 $m$ 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 $\hat{f}$, parametrized by $\hat{\theta}$ with output function $\hat{h}_0$. 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 $\hat{f}$ so that

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

or

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

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):

$$\begin{cases} x_{k+1} = f_0(x_k) \\ y_k = h_0(x_k) + m_k \end{cases} \quad (3)$$

The generic equations of the receiver read:

$$\begin{cases} \hat{x}_{k+1} = \tilde{f}_0(\hat{x}_k, y_k) \\ \hat{y}_k = \tilde{h}_0(\hat{x}_k) \end{cases} \quad (4)$$

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...
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 \( f^i_0 \) with output function \( h^i_0 \) \((i = 1, \ldots, N) \). Therefore, in the transmitter description, the index \( i \) depends on \( m_k \).

\[
\begin{align*}
x_{k+1} &= f^{\hat{m}_k}_i(x_k) \\
y_k &= h^{\hat{m}_k}_i(x_k) .
\end{align*}
\]  

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).

![Chaotic Switching](image)

**Fig. 3. Chaotic Switching**

The objective at the receiver end is to decide which chaotic system \( f^i_0 \) is most likely to have produced the sequence \( \{y_k\}_{jK,(j+1)K-1} \). To this end, the receiver part is composed of as many systems, say \( N \), as at the transmitter part:

\[
\begin{align*}
\hat{x}_{k+1} &= f^i_0(\hat{x}_k, [y_k]) \\
\hat{y}_k &= h^i_0(\hat{x}_k) .
\end{align*}
\]  

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 \( f^i_0 \) 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) \((\text{where } x_0 \text{ must be replaced by } x_{jK}) \) is obtained by unidirectional coupling through the variable \( y_k \) which is really involved in \( f^i_0 \) 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 \( r^i_k \), where

\[ r^i_k = h^{\hat{m}_k}_0(x_k) - h^i_0(\hat{x}_k) .\]

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 \( \lambda \) \((\text{different from the key parameter } \theta) \) 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:

\[
\begin{align*}
x_{k+1} &= f^{\hat{m}_k}_0(x_k) \\
y_k &= h^{\hat{m}_k}_0(y_k) .
\end{align*}
\]  

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 \( y_k \):

\[
\begin{align*}
\hat{x}_{k+1} &= f^i_0(\hat{x}_k, y_k) \\
\hat{y}_k &= h^i_0(\hat{x}_k) .
\end{align*}
\]  

Only one observer, set with the same value \( \lambda^i \) of the transmitter which has actually delivered the sequence \( \{y_k\}_{jK,(j+1)K-1} \), can be synchronized in the form (1) or (2) \((\text{where } x_0 \text{ must be replaced by } x_{jK}) \) 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

\[ r^i_k = h^{\hat{m}_k}_0(x_k) - h^i_0(\hat{x}_k) .\]

For the continuous modulation (Fig. 4b), the information \( m_k \) takes values over an uncountable set \( M \). Consequently, an infinite number of units at the receiver side would be required. As a matter of fact, for the recovering of \( x_k \) and of \( m_k \), we usually resort to adaptive techniques and identification procedures [10][6][14][2]. The estimation must fulfill \( \hat{x}_k = x_k \) 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 \( \lambda^{-1}(\lambda^i) \) for the continuous modulation, leads to a unique value \( m_k \).
Nevertheless, for the parameter modulation and similarly to the chaotic switching, the information must be constant during a prescribed switching 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].

Fig. 4. Parameter modulation

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 mk is the required number r of iterations of the output yk so as yk+r depends on mk which actually appears explicitly in the expression of yk+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 xk, the output yk+r reads

\[ y_{k+r} = h_0(f_0(x_k, m_k)) \]

where

\[ f_0(x_k, m_k) = x_k \text{ when } i = 0 \]
\[ = f_0(f_0^{-1}(x_k, m_k), m_{k+i-1}) \forall i \geq 1. \]

and where mk appears explicitly in the sense that there exists \( m_k' \neq m_k \) such that \( y_{k+r} = h_0(f_0'(x_k, m_k)) \) whereas for all \( m_k' \neq m_k \), \( y_{k+r} = h_0(f_0'(x_k, m_k)) = h_0(f_0'(x_k, m_k')) \) 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 \( r \geq 0 \):

\[ \begin{cases} \hat{x}_{k+1} = f_0(\hat{x}_k, y_k, \ldots, y_{k+r}) \\ \hat{m}_k = g(\hat{x}_k, y_{k+r}) \end{cases} \]

with \( g \) such that

\[ \hat{m}_k = g(\hat{x}_k, y_{k+r}) = m_k \text{ when } \hat{x}_k = x_k. \]

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
The quicker the brute force attack succeeds, the cryptosystem plays a central role because it is intended to recover is of paramount importance. Amazingly, only a somewhat limited interest has been attached so far to this parameter vector (or equivalently the system) is structurally identifiable. On the other hand, it is computationally easy for polynomial systems to obtain (17). Retrieving the parameter in this way constitutes an algebraic attack, meaning that if there exists an open neighborhood of \( \theta \), \( \Theta \subset \Theta \), and some dense subsets \( v(\theta) \subset \Theta \) and \( M_0^\Theta \subset M \), 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 \( x_0 \in X, \theta \in v(\theta) \) and \( \{m_k\}_0^T \in M_0^\Theta \). 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

\[
\Phi(m_k, y_k; \theta) = 0,
\]

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 \( i = 1, \ldots, L \),

\[
P_i(y_k, \ldots, y_{k+N}, m_k, \ldots, m_{k+N}) \theta^{(i)} - Q_i(y_k, \ldots, y_{k+N}, m_k, \ldots, m_{k+N}) = 0, \quad i = 1, \ldots, L,
\]

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

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

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. The 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 parameter identifiability.

**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 \( \theta \), \( \nu(\theta) \subset \Theta \), such that for any \( \hat{\theta} \in \nu(\theta) \) and for any \( \theta \in v(\theta) \):

\[
\hat{\theta} \neq \theta \Rightarrow \{y_k(x_0, m_k, \hat{\theta})\}_0^T \neq \{y_k(x_0, m_k, \theta)\}_0^T
\]

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 \( M_0^\Theta \subset M \), 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 \( \{m_k\}_0^T \in M_0^\Theta \).
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 $f_0$ and output function $h_0$. Besides, a function $v_c$, depending on a time-varying quantity, say, the state vector $x_k$ of the chaotic system, encrypts the information $m_k$ and delivers $u_k = v_c(x_k, m_k)$. Then, the signal $u_k$ is transmitted via a second channel. The set of equations governing the transmitter is

$$\begin{align*}
x_{k+1} &= f_0(x_k), \\
y_k &= h_0(x_k), \\
u_k &= v_c(x_k, m_k),
\end{align*}$$

(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:

$$\begin{align*}
\hat{x}_{k+1} &= \hat{f}_0(\hat{x}_k, y_k) \\
\hat{y}_k &= \hat{h}_0(\hat{x}_k), \\
\hat{m}_k &= v_d(\hat{x}_k, u_k)
\end{align*}$$

(19)

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

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

(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.

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 $v_d$, 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

$$\begin{align*}
\hat{x}_{k+1} &= \hat{f}_0(\hat{x}_k, y_k, \ldots, y_{k+i}), \\
\hat{u}_k &= g_0(\hat{x}_k, y_{k+i}), \\
\hat{m}_k &= v_d(\hat{x}_k, \hat{u}_k)
\end{align*}$$

(24)

with $g$ such that

$$\hat{u}_k = g_0(\hat{x}_k, y_{k+i}) = u_k \text{ when } \hat{x}_k = x_k.$$  

(25)

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, $u_k = v_c(x_k, m_k)$ is not directly conveyed through the channel but is reinserted into the chaotic dynamics $f_0$ (and possibly into $h_0$). 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 $v_c(x_k, m_k)$. Conversely, when $v_c$ is such that $v_c(x_k, m_k) = m_k$ 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 $u_k$:

$$\begin{align*}
x_{k+1} &= f_0(x_k, u_k) \\
y_k &= h_0(x_k, u_k) \\
u_k &= v_c(x_k, m_k)
\end{align*}$$

(21)

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

$$\begin{align*}
x_{k+1} &= f_0(x_k, u_k) \\
y_k &= h'_0(x_k) \\
u_k &= v_c(x_k, m_k)
\end{align*}$$

(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

$$y_{k+r} = h'_0(f_0^r(x_k, u_k)),$$

(23)

where

$$f_0^r(x_k, u_k) = x_k \text{ when } i = 0 = f_0(f_0^{i-1}(x_k, u_k), u_{k+i-1}) \forall i \geq 1.$$

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 $v_d$, 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

$$\begin{align*}
\hat{x}_{k+1} &= \hat{f}_0(\hat{x}_k, y_k, \ldots, y_{k+i}) \\
\hat{u}_k &= g_0(\hat{x}_k, y_{k+i}) \\
\hat{m}_k &= v_d(\hat{x}_k, \hat{u}_k)
\end{align*}$$

(24)

with $g$ such that

$$\hat{u}_k = g_0(\hat{x}_k, y_{k+i}) = u_k \text{ when } \hat{x}_k = x_k.$$  

(25)
and with $v_d$ such that
\[ \hat{m}_k = v_d(\hat{x}_k, \hat{u}_k) = m_k \text{ when } \hat{x}_k = x_k \text{ and } \hat{u}_k = u_k. \]  
(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 $v_e$, possibly non-polynomial, which can make the state generator significantly more resistant to algebraic attacks, as described in Subsection II-C.2.

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

\[
\begin{align*}
K_{k+1} &= \sigma_\theta^s(K_k) \\
\hat{c}_k &= e(K_k, m_k)
\end{align*}
\]

(27)

The key \(K_k\) is generated by a function \(\sigma_\theta^s\) parameterized by \(\theta\), the parameter \(\theta\) acting as the secret static (or master) key. The ciphertext \(\hat{c}_k\) 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

\[
\begin{align*}
K_{k+1} &= \sigma_\theta^s(c_{k-l}, \ldots, c_{k-r}) \\
\hat{c}_k &= e(K_k, m_k)
\end{align*}
\]

(28)

where \(\sigma_\theta^s\) 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

\[
\begin{align*}
\hat{K}_{k+1} &= \sigma_\theta^s(\hat{K}_k) \\
\hat{m}_k &= d(\hat{K}_k, \hat{c}_k)
\end{align*}
\]

(29)

and, in the SSSC case, by

\[
\begin{align*}
\hat{K}_{k+1} &= \sigma_\theta^s(c_{k-l}, \ldots, c_{k-r}) \\
\hat{m}_k &= d(\hat{K}_k, \hat{c}_k)
\end{align*}
\]

(30)

In both cases, the decryption function \(d\) is such that

\[
\hat{m}_k = d(\hat{K}_k, \hat{c}_k) = m_k \text{ when } \hat{K}_k = K_k.
\]

(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 \(\sigma_\theta^s\) 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 \(\sigma_\theta^s\). If the parameters are identical at both sides, the respective keystreams synchronize automatically because \(\sigma_\theta^s\) 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. \(\hat{c}_k\)). 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
text)

Definition 5: (Flatness) A system with dynamic $f$,
parametrized by $\theta$, of relative degree $r$, input $\epsilon_k$ 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 $\mathcal{T}_\theta$ and $\mathcal{G}_\theta$ which obey

$$
\begin{align*}
  z_k &= \mathcal{T}_\theta(y_k; x_k, u_k) \\
  \epsilon_k &= \mathcal{G}_\theta(y_k; x_k, u_k)
\end{align*}
$$

(32)

where $k_F(r), k'_F(r), k_G(r)$ and $k'_{G}(r)$ 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 $u_k$ 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 $\mathcal{T}_\theta$
and $\mathcal{G}_\theta$ and integers $k_F(0), k'_F(0), k_G(0)$ and $k'_{G}(0)$ such that

$$
\begin{align*}
  x_k &= \mathcal{T}_\theta(y_k; k_F(0), k'_F(0)) \\
  m_k &= \mathcal{G}_\theta(y_k; k_F(0), k'_F(0))
\end{align*}
$$

(33)

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

$$
\begin{align*}
  x_{k+1} &= \mathcal{T}_\theta(y_{k+1}; k_F(0), k'_F(0)) \\
  y_k &= \mathcal{G}_\theta(y_{k+1}; k_F(0), k'_F(0)) + 1
\end{align*}
$$

(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 $\mathcal{T}_\theta = \mathcal{T}_\theta$, running key $K_k = x_k$, ciphertext
$c_k = y_k$, encrypting function $e = h$, secret static key $\theta$ and
memory $b = |k_F(0) - k'_F(0)| + 1$.

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

$$
x_{k+1} = f_\theta(\mathcal{F}_\theta(y_k; k_F(0), k'_F(0)), \mathcal{G}_\theta(y_k; k_F(0), k'_F(0)))
$$

(35)

Besides, according to Definition 5, flatness of (10), with
relative degree $r > 0$, means that there exist two functions $\mathcal{T}_\theta$
and $\mathcal{G}_\theta$ and integers $k_F(r), k'_F(r), k_G(r)$ and $k'_{G}(r)$ such that

$$
\begin{align*}
  x_k &= \mathcal{T}_\theta(y_k; k_F(r), k'_F(r)) \\
  m_k &= \mathcal{G}_\theta(y_k; k_F(r), k'_F(r))
\end{align*}
$$

(36)

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

$$
\begin{align*}
  x_{k+1} &= \mathcal{T}_\theta(y_{k+1}; k_F(r), k'_F(r)) \\
  y_{k+1} &= \mathcal{T}_\theta(y_{k+1}; k_F(r), k'_F(r)) + 1
\end{align*}
$$

(37)

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

$$
\begin{align*}
  x_{k+1} &= \mathcal{T}_\theta(y_{k+1}; k_F(r), k'_F(r)) \\
  y_{k+1} &= \mathcal{T}_\theta(y_{k+1}; k_F(r), k'_F(r)) + 1
\end{align*}
$$

(38)

where $r > 0$, 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 $k_F(r), k'_F(r), k_G(r)$ and $k'_{G}(r)$ and writing down
explicitly the functions $\mathcal{T}_\theta$ and $\mathcal{G}_\theta$. It can be shown (see [7]
for the linear case) that resorting to a state reconstruction
algorithm 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 $\mathcal{T}_\theta$ 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 Defini-
tion 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 $\mathcal{T}_\theta$ and $\mathcal{G}_\theta$ and integers $k_F(0), k'_F(0), k_G(0)$ and $k'_{G}(0)$ such that

$$
\begin{align*}
  x_k &= \mathcal{T}_\theta(y_k; k_F(0), k'_F(0)) \\
  u_k &= \mathcal{G}_\theta(y_k; k_F(0), k'_F(0))
\end{align*}
$$

(39)

When iterating once forward the first equation of (39), it
turns out that (21) is strictly equivalent to
depends explicitly on \( x_k \) and \( m_k \), identification of (40) with (28) leads then to the following result:

\[
\begin{align*}
  x_{k+1} &= \mathcal{F}_\theta(y_k + k_{r}(x_k, m_k) + y_{k+1} + k'_{r}(y_k + m_k)) \\
  y_k &= h_\theta(x_k, v_e(x_k, m_k))
\end{align*}
\] (40)

Letting \( I_{k,r}(x_k, m_k) = h_\theta(x_k, v_e(x_k, m_k)) \) since \( y_k \) depends explicitly on \( x_k \) and \( m_k \), identification of (40) with (28) leads then to the following result:

\[
\begin{align*}
  x_{k+1} &= \mathcal{F}_\theta(y_k + k_{r}(x_k, m_k) + y_{k+1} + k'_{r}(y_k + m_k)) \\
  y_k &= h_\theta(x_k, v_e(x_k, m_k))
\end{align*}
\] (40)

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

\[
\begin{align*}
  x_k &= \mathcal{F}_\theta(y_k + k_{r}(x_k, m_k) + y_{k+1} + k'_{r}(y_k + m_k)) \\
  u_k &= G_\theta(y_k + k_{r}(x_k, m_k) + y_{k+1} + k'_{r}(y_k + m_k))
\end{align*}
\] (41)

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

\[
\begin{align*}
  x_{k+1} &= \mathcal{F}_\theta(y_k + k_{r}(x_k, m_k) + y_{k+1} + k'_{r}(y_k + m_k)) \\
  y_{k+r} &= h_\theta(x_k, v_e(x_k, m_k))
\end{align*}
\] (42)

Letting \( I_{r,f,v}(x_k, m_k) = h_\theta(x_k, v_e(x_k, m_k)) \) since \( y_{k+r} \) depends explicitly on \( x_k \) and \( m_k \), identification of (42) with (28) leads then to the following result:

\[
\begin{align*}
  x_{k+1} &= \mathcal{F}_\theta(y_k + k_{r}(x_k, m_k) + y_{k+1} + k'_{r}(y_k + m_k)) \\
  y_{k+r} &= h_\theta(x_k, v_e(x_k, m_k))
\end{align*}
\] (42)

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

\[
m_k = v_e(\mathcal{F}_\theta(y_k + k_{r}(x_k, m_k) + y_{k+1} + k'_{r}(y_k + m_k)), G_\theta(y_k + k_{r}(x_k, m_k) + y_{k+1} + k'_{r}(y_k + m_k)))
\] (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 \( v_e \), 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:

\[
\begin{align*}
  x_{k+1} &= A x_k + B u_k \\
  y_k &= C x_k \\
  u_k &= v_e(x_k, m_k)
\end{align*}
\] (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

\[
A = \begin{bmatrix} 38 & 1 & 0 \\ 7 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.
\]

It is recalled that, for linear systems written in a state space form, the relative degree corresponds to the smallest integer \( r \) such that \( CA^{-1}B \) 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 \( v_e \) 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

\[
\begin{align*}
  x_{k+1} &= = y_k \\
  y_{k+1} &= 7y_k - 7y_{k-1} + 4y_{k-2} \\
  y_k &= 4y_{k-1}
\end{align*}
\] (45)

and the function \( G_\theta \) obeying

\[
\begin{align*}
  u_k &= y_{k+1} - 38y_k - 7y_{k-1} - 4y_{k-2} .
\end{align*}
\] (46)

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

\[
\begin{align*}
  y_{k+1} &= CA x_k + CB v_e(x_k, m_k) \\
  &= 38x_k(1) + x_k(2) \oplus x_k(3) \oplus m_k + x_k(2) \oplus x_k(3) \oplus m_k
\end{align*}
\] (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 \( \sigma_{\theta} = \mathcal{F}_\theta \) corresponding to Eq. (45), running key \( K_r = x_k \), ciphertext \( c_t = y_{k+1} \), encrypting function \( e = I_{r,f,v} \) 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 $v_d$ is also an XOR between the components of $x_k$ and the (pre-)ciphertext $u_k = v_r(x_k, m_k)$, that is, $v_d(x_k, u_k) = u_k \oplus x_k^{(1)} \oplus x_k^{(2)} \oplus x_k^{(3)}$ where $x_k^{(i)}$ and $u_k$ are meant to be the corresponding 8-bit representation similarly to the function $v_r$. 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 $\mathcal{F}_p$.

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 $m_k$ ($v_r$ and so $v_d$ 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 $v_r$, which may constitute a challenging task for further research.

**Acknowledgment** Jos’e Maria Amigó was partially funded by the Spanish Ministry 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**


