Physics Letters A 377 (2013) 760–765

Contents lists available at SciVerse ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

Secure information transfer based on computing reservoir

R.M. Szmoski [a], F.A.S. Ferrari [a], S.E. de S. Pinto [a,*], M.S. Baptista [b], R.L. Viana [c]

[a] Department of Physics, Universidade Estadual de Ponta Grossa, 84030-900, Ponta Grossa, Brazil
[b] Institute for Complex Systems and Mathematical Biology, SUPA, University of Aberdeen, Aberdeen, United Kingdom
[c] Department of Physics, Universidade Federal do Parana, 81531-990, Curitiba, Parana, Brazil

ARTICLE INFO

Article history:
Received 9 August 2012
Accepted 15 January 2013
Available online 18 January 2013
Communicated by A.P. Fordy

Keywords:
Chaotic coupled maps
Computing reservoir
Information transmission
Networks
Transfer entropy

ABSTRACT

There is a broad area of research to ensure that information is transmitted securely. Within this scope, chaos-based cryptography takes a prominent role due to its nonlinear properties. Using these properties, we propose a secure mechanism for transmitting data that relies on chaotic networks. We use a nonlinear on–off device to cipher the message, and the transfer entropy to retrieve it. We analyze the system capability for sending messages, and we obtain expressions for the operating time. We demonstrate the system efficiency for a wide range of parameters. We find similarities between our method and the reservoir computing.

1. Introduction

A communication system is a device whose purpose is to convey a message between spatially distinct locations. Schematically, there are five elements in any communication system, namely: an emitter, an encoder, a transmission channel, a decoder, and a receiver [1]. The encoder handles the message, so it passes through the channel, and the decoder carries out the reverse process. In order to be practical and safe, the construction of a communication system may require other components [2].

A major concern of communication is the security in data transmission. In many cases, the speed and reliability for transmitting a message with low probability of errors is not enough, but also the transmission has to be carried out in an extremely secure way. In 1949, C.E. Shannon took a decisive step toward showing that if the length of the key is not an inconvenience, a message can be securely sent [1]. In a communication system, traditional cryptography functions in the software level rather than in the high speed physical level. Besides, security requires the use of ergodic and mixing transformations also in the software level. In the early 1990s an attempt was made to show how to use chaos synchronization to create a secure communications systems [3,4]. This kind of synchronization has enabled one to create a fast cryptographic system that operates in the physical (hardware) level of the communication system. Besides, chaotic systems are naturally ergodic and mixing, which provides security. This initial idea was shown to be insecure [5]; however, the complexity of nonlinear systems enables other approaches. On account of this, different communication systems based on chaos synchronization have been proposed and investigated [6,7]. In general, such systems assume chaotic dynamics for both the decoder and receiver. Thus, the encoder codifies a message using some property of the chaotic signal and, after being transmitted, the message is decoded by the receiver, which is synchronized to the emitter [8].

An eavesdropper trying to determine the parameters of a cryptographic system will always make an error. If the dynamics of the system is chaotic, a small error grows exponentially what makes difficult to decrypt the intercepted message [9–14], if the decoding relies on a receiver that is exactly identical to an emitter. This is one of the properties that made chaos-based cryptosystems popular.

Synchronization offers a communication system where information is transmitted and received in real time and that operates in physical level. However, the need for synchronization reduces the parameter range of the transmitter and receiver within which the system can be considered secure. For example, in [15] it was shown that receiver and emitter do not need to match for the retrieval of information.

A different approach to secure communication using coupled chaotic systems was presented by Hung and Hu [16]. In their method, a binary message is codified considering the coupling direction between chaotic maps on a ring. The receiver decodes the message, determining the transfer entropy [17] between

* Corresponding author.
E-mail addresses: desouzapinto@pq.cnpq.br, desouzapinto@gmail.com (S.E. de S. Pinto).

0375-9601/$ – see front matter
http://dx.doi.org/10.1016/j.physleta.2013.01.021

succeeding maps. Indeed, for any interacting system, the transfer entropy can be used for determining which variables influence the dynamics of each other. The novel feature of this method is that it does not require synchronization for transmitting data; it only requires the determination of the transfer entropy. One inconvenience is that, as this quantity is statistically defined, many observations from the dynamics of the maps are necessary in order to decode the message. The observation interval needed by the receiver to determine the transfer entropy between the maps was taken as the relaxation time [16]. Therefore, the transmission of information through this mechanism requires operating times that are multiples of the relaxation time.

In this work, we propose an improvement over the previous method of Hung and Hu consisting of a communication system that uses both chaos synchronization as well as transfer entropy. While Hung and Hu's method uses only one network of coupled maps, our model contains two networks, the emitter and the receiver, the latter being a replica of the former. The transmission of a message through the system is accomplished by two stages that we call pre- and post-synchronization. Here, synchronization occurs just between the elements with the same label, in the emitting and receiving networks. Importantly, the elements in each network are out of synchronization. The use of desynchronized networks allows one to explore the good properties of synchronization, namely fast transmission of information, but without losing the parameter range for which the system is secure [18]. Mostly important, since that all nodes in the emitter network are desynchronous, the encoded signal generated by this network is composed by variables that are not correlated. In our method, it is strictly necessary that receiver is identical to emitter in order to decode the message. Since the transmitter is composed by a large network of desynchronous elements, it is very difficult to determine the parameters of this network by analyzing the transmitted signal. An eavesdropper cannot reproduce equally the emitter. Even if an intruder manages to find all values of the state variables from the emitter and receiver, if one does not know precisely the coupling function and intensity, one is not capable of decoding the message correctly. Assuming the coupling prescription to be secret keys of the communication system, if the key is altered after each communication, then security would be increased further.

In the stage of pre-synchronization, the networks start interacting directly with each other until every node of the emitting network becomes identical to other every node with the same label of the receiving network. Once the emitter and receiver networks are synchronized, the second stage starts transmitting and decoding the message. During this stage, the interaction between networks is deactivated and the message is encoded in a binary signal through an on–off device. The advantage of this procedure consists on transmitting N bits of information for each unit of the relaxation time. In other words, the communication device transmits N bits of information using a bit stream, which makes data transmission fast.

The proposed method relies on the fact that a long scalar quantity, composed by the trajectory of a uni-dimensional system (s) coupled in a master–slave fashion with an N-dimensional emitter network by a connecting matrix representing the binary message to be transmitted, carries information about these couplings and, therefore, the message. The message can only be decoded by a person that has complete knowledge of the emitter network. Imagine the system s as a node in a large dynamical network. With the exception of the node s, assume that the information about all other nodes is either known or can be precisely measured. Our method works because it is possible to determine, which are the nodes in this dynamical network that are connected to s by measuring the flow of information created by a connection. Hence, nodes coupled to s influence its dynamical behavior. Furthermore, the receiver uses the transfer entropy to decode the message.

The fundamentals behind the success of our cryptographic method share similarities with one possible way in which the information is believed to be processed and transmitted in the brain. Reservoir Computing (RC) [19] is a machine-learning paradigm employed to retrieve from a dynamical network, the reservoir (the "brain"), information of an external perturbation driving it, input. The assumption behind RC is that the information about the input is spread out all over the network, and reliable retrieval of it can be accomplished by making a weighted average from the trajectories of some selected nodes of the reservoir, the output. Discovering which nodes should be selected is a remarkable task that can be resolved by a learning process whose purpose is to find an approximate match between input and output. In summary, one hopes to find the connecting topology between the reservoir and output (for a given random connecting topology between the input and the reservoir), such that the output matches the input. It has been recently demonstrated that RC can be performed by a reservoir composed by a single dynamical node operating as if it were a complex system [20]. In order to make the analogy between RC and the proposed cryptographic method, imagine the system s functioning as the reservoir, the emitter network being responsible to produce the input, the connecting topology between the input and s is given by the message, and the output is generated by the receiver network, which in our method equals the input in the post-synchronization stage. Similarly, to RC, which considers that the input perturbs any random set of nodes in the reservoir, the proposed cryptographic method works regardless of how is the connecting topology between the input and s, the message. Any random message can be decoded by whom has confidential information. The main assumption for the success of RC lies behind the belief that the reservoir stores information about the perturbation experienced by it. According to this analogy, we argue that the state of the reservoir stores specific information about the way how it is disturbed. In contrast to RC, that aims at discovering the connecting topology between the reservoir and output, in the proposed cryptographic method it is the connecting topology between input and the reservoir, the message that needs to be revealed. Finally, in RC input and output only roughly match, whilst in our method, they need to match perfectly; otherwise, the message cannot be decoded. Therefore, this analogy allows one to interpret the encoding system s as a sort of reservoir that has memory about the message, i.e., the way it is being perturbed.

This work is organized as follows: in Section 2, we describe the communication device based on both chaos synchronization and transfer entropy analysis. Section 3 describes the pre-synchronization stage, computing the synchronization time of a lattice of coupled piecewise-linear chaotic maps. In Section 4 we describe the post-synchronization stage, determining the relaxation time of the system. Section 5 examines an example of data transmission through this mechanism. The last section is devoted to our conclusions.

2. Description of the communication mechanism

Consider the emitter (E) and receiver (R) networks as two identical coupled map lattices composed by N sites each one [21–23]. The number of sites N is defined according to the amount of bits in a message that will be transmitted. If we assume the message as a binary sequence $m = \{m^{(1)}m^{(2)}, \ldots, m^{(N')}\}$ of length N', with $m^{(i)}$ equal to 0 or 1, N is equal to the number of elements belonging to this sequence, this is $N = N'$. The state of the E and R networks, at each discrete time n, is defined by the vectors $\mathbf{e}_n = (e_n^{(1)}, e_n^{(2)}, \ldots, e_n^{(N)})^T$ and $\mathbf{r}_n = (r_n^{(1)}, r_n^{(2)}, \ldots, r_n^{(N)})^T$, respectively, with $e_n^{(i)}$ and $r_n^{(i)}$ corresponding to a state variable $z \in \Omega$ whose time evolution is governed by a chaotic map $f : \Omega \mapsto \Omega$, with $\Omega \subset \mathbb{R}$.

There have been investigations of chaos synchronization between replicas of coupled map networks using continuous maps [24–26]. Based on such previous investigations we can restrict our analysis to continuous maps f over the set Ω. In particular, we focus on the piecewise-linear tent map $f(z) = 1 - 2|z - 0.5|$ [27].

Besides the individual dynamics, the sites in each network are submitted to a coupling prescription. This intra-network coupling is arbitrary, but, for transmitting and sending a message correctly both E and R networks must be taken as identical, sharing the same coupling prescription and parameters. So, our communication system uses a symmetric private key. Here we consider a Laplacian-local intra-network coupling with periodic boundary conditions and random initial conditions:

$$F\big(z^{(i)}\big) = (1-\varepsilon) f\big(z_n^{(i)}\big) + \frac{\varepsilon}{2}\big[f\big(z_n^{(i-1)}\big) + f\big(z_n^{(i+1)}\big)\big], \quad (1)$$

where $z^{(i)} = z^{(N \pm i)}$ represents the state variable of the i-th site ($i = 1, \ldots, N$) and $\varepsilon \in [0, 1]$ stands for the strength of intra-network coupling in each network.

The transmission data process between E and R networks is composed by two stages: the pre- and post-synchronization. When we refer to synchronization we mean the process in which $\mathbf{e}_n = \mathbf{r}_n$ for all n. The components of each state vector are necessarily not equal. If all maps of a network mutually synchronize, then the dynamics of the network, given by Eq. (1), reduces to the dynamics of an uncoupled map. This is an undesirable feature for the point of view of the secure communication, since an intruder could determine the network state from knowing the state of only one site, that would endanger the security of the transmission. Thus, in order to avoid mutual synchronization in each network, we will assume that intra-network coupling intensity is sufficiently weak, which increases the dimension of the emitter.

3. Pre-synchronization stage

A way of synchronizing the state vectors of the E and R networks is to assume an interaction between them. This inter-network coupling may be unidirectional or bidirectional. In the first case, also known as master–slave coupling, one network influences the dynamics of the other but is not influenced by the latter; while, in the second case, both networks influence and are influenced by each other. We will take here the master–slave coupling, E as the master and R as the slave networks, such that the dynamics of the system is described by

$$e_{n+1}^{(i)} = F\big(e_n^{(i)}\big),$$
$$r_{n+1}^{(i)} = (1-\gamma) F\big(r_n^{(i)}\big) + \gamma F\big(e_n^{(i)}\big), \quad (2)$$

in which γ is the strength of the inter-network coupling. Fig. 1 illustrates the system described by Eqs. (1) and (2). Each site is represented by a disc and the lines indicate the connections among them. Note that the E-sites interact with their neighbors inside the E-network, and R-sites interact with their neighbors inside the R-network (intra-network connections are bidirectional). However, the sites of E influence a corresponding site of R and its nearest neighbors, since the inter-network connections are unidirectional.

The coupling remains active while the networks do not synchronize with each other. When this occurs the network E stops sending information about its state variables. Since the synchronization of the E and R networks marks the end of the first stage of the process, we have to identify when it occurs. We use as a synchronization diagnostic, the synchronization error average, given by

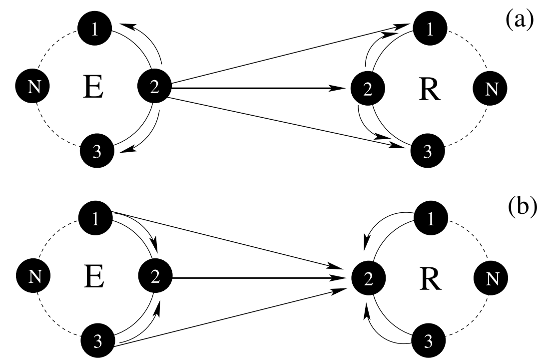

Fig. 1. Schematic diagram of the connections between corresponding sites (labeled 2) of two networks (E: emitter; R: receiver) with Laplacian-local intra-network coupling and a master–slave inter-network coupling. (a) Site 2 in both networks have outcoming intra-network coupling. Outcoming inter-network coupling is described by site 2 of E coupled to sites 1, 2, and 3 of R; (b) Site 2 in both networks have incoming intra-network coupling. Incoming inter-network coupling is described by site 2 of R coupled to sites 1, 2, and 3 of E.

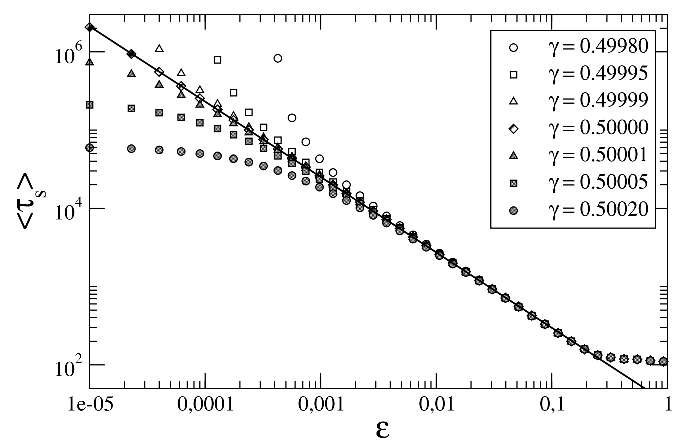

Fig. 2. Average synchronization time as a function of the intra-network coupling parameter ε for different values of the inter-network coupling parameter γ. The solid line is a least squares fit with slope -1.0.

$$w_n = \frac{1}{N}\sum_{i=1}^{N}\big|e_n^{(i)} - r_n^{(i)}\big|. \quad (3)$$

If the state vectors of E and R networks are synchronous we have $w_n = 0$, otherwise $w_n > 0$. The synchronization time τ_s is the time it takes for this average error to be less than 10^{-14} over a time window of 1000 consecutive iterations of the system.

In order to make the $\mathbf{e}_n$ and $\mathbf{r}_n$ vectors identical we need to choose the γ and ε parameters in such a way that they allow synchronization. In the particular case in which $\varepsilon = 0$, synchronization is just observed when $\gamma > \gamma_c = 1/2$. For an inter-network coupling strength of $\gamma = 1/2 \pm \delta$, we investigated the synchronization time τ_s between the networks as a function of the intra-network coupling strength ε. Fig. 2 shows the average synchronization time as a function of the intra-network coupling strength for 100 different randomly chosen initial conditions, $N = 21$ and for different values of δ. We see that, for $\gamma = 0.5$ ($\delta = 0$) and $\varepsilon < 0.3$, the average synchronization time scales with ε as a power-law

$$\langle \tau_s \rangle = C\varepsilon^{-\alpha}, \quad (4)$$

in which C and α are obtained by the best fit of the points. In the case presented we have the following values: $\alpha = 1.018$ and $C = 32.23$. Considering networks of different sizes, we found that $\alpha \approx 1.0$ while C depends on the value of w from which the networks are considered synchronized, more specifically, $C = \ln w^{-1}$. Even though this relation, in general, is not a power-law for any

Table 1
XOR true table.

Message	0	0	1	1
Keystream	0	1	0	1
XOR	0	1	1	0

Table 2
Cipher and decipher operations.

Plain Text	$\oplus$	Keystream	=	ciphertext
ciphertext	$\oplus$	keystream	=	Plain text

value of ε, it can be used nevertheless to estimate the synchronization time between the networks. In the following we will consider just the case for $\varepsilon \leqslant 0.1$ and, thus, Eq. (4) allows us to estimate the time it takes for the coupled networks to mutually synchronize.

4. Post-synchronization stage

In the previous section, we showed how to synchronize the networks E and R by controlling the inter-network coupling parameter γ. Here, we will describe how to convey a particular message m between E and R in a safe and efficient way. A necessary condition is that both networks remain synchronized during all the process, otherwise the receiver will not be able to read the message correctly.

Remember that we consider the networks to be synchronized whenever the synchronization error w_n is less than a tolerance fixed at 10^{-14}. In order to keep the network synchronized we truncate the state variables of both E and R such that we get rid of differences $o(w)$ less than 10^{-14}. Then we turn off the inter-network coupling, since it is no longer necessary for keeping the networks synchronized. Indeed, E and R being identical networks, if $\mathbf{e}_n = \mathbf{r}_n$ at a given instant $n = \tau_s$ then they remain synchronized for all $n > \tau_s$.

After the inter-network coupling is switched off the second stage of the communication process begins, in which we transmit the desired message. To compare our method with traditional cryptography, we briefly introduce the famous Vernam cipher [28], a symmetrical key cipher where plain text digits are combined with a keystream. This combination produces a ciphertext using the operation XOR, symbolized by $\oplus$, whose true table is presented in Table 1. The operation is reciprocal: one uses an identical keystream both to encipher plain text to ciphertext and to decipher ciphertext to yield the original plain text (Table 2).

According to Shannon, for a cipher to be considered secure it must satisfy the following conditions: (i) the keystream must have at least the same length of the message, (ii) is changed at every communication, and (iii) binary symbols of keystream must be randomly decorrelated, then cipher is proven to be secure [2].

In our method, the XOR transformation is a sophisticated nonlinear transformation. Similarly to the Vernam cipher, the size of the emitter network is equal to the length of the message. The emitter network has a role similar to the keystream in the Vernam cipher method. Finally, the requirement that a keystream must have decorrelated symbols is analogously reproduced in our method by having decorrelated nodes in the emitter network.

We introduce a discrete-time dynamical system $S : \Omega \mapsto \Omega$ that is responsible for encoding the message in the signal consisting of an orbit of the system S. The map S defines the value of the signal s_n in each time instant associating the message characters to the state variables of the emitter network through an on–off device. If the i-th element of the message m, denoted as $m^{(i)}$, has a binary value of 1, then $e_n^{(i)}$ influences the dynamics of s_n (mode-on), whereas if $m^{(i)} = 0$, then $e_n^{(i)}$ does not influence the signal dynamics (mode-off). Moreover, the dynamics of the signal is given by the following map

$$s_{n+1} = S(s_n) = (1-\beta)s_n + \frac{\beta}{\eta}\sum_{i=1}^{N} e_n^{(i)} m^{(i)}, \tag{5}$$

in which $\beta \equiv (N-1)/N$ and $\eta = \sum_{i=1}^{N} m^{(i)}$ is a normalization factor. Given a randomly chosen initial condition s_0 the iteration of the map S yields a chaotic orbit $\{s_n\}_{n=0}$. The map S itself is not chaotic but, since it is driven by $e_n^{(i)}$, that it is itself chaotic, the orbit $\{s_n\}_{n=0}$ results chaotic as well.

Notice that the magnitude of the signal is not a feature of the message, it is rather a feature of the dynamics of each element of the emitter network (E). Consequently, if the system initiates from different initial states, the same message will generally result in different signals. On the other hand, it may happen that two different messages result in the same signal for a limited and usually short period of time, however the signals will eventually diverge with time.

For recovering the message contained in the signal $\{s_n\}_{n=0}$ the receiver network R first verifies which state variables $e_n^{(i)} = r_n^{(i)}$ influence and which do not influence the s_{n+1}. Then the receiver network associates the symbol 1 to the former case and 0 to the latter case, observing the indexes of the sites in the network. The transfer entropy is the dynamical tool that allows the receiver network R to accomplish such verification. The transfer entropy $T_{r_n^{(i)} \to s_{n+1}}$ vanishes if and only if the dynamics of s_{n+1} does not depend on the dynamics of $r_n^{(i)}$, so if the transfer entropy is nonzero there is a statistical coherence between these signals. It is defined as

$$T_{r_n^{(i)} \to s_{n+1}} = \sum p\left(s_{n+1}, s_n, r_n^{(i)}\right) \log \frac{p(s_{n+1}|s_n, r_n^{(i)})}{p(s_{n+1}|s_n)}, \tag{6}$$

where $p(\ ,\ ,\ )$ and $p(\ |\ )$ mean the join and conditional probabilities, respectively. These probabilities may be calculated using a box counting algorithm or a kernel estimator [17]. In this work we use the former procedure by considering the following coarse-grained variables

$$\tilde{x} = \begin{cases} 0, & \text{if } 0 < x \leqslant 1/2, \\ 1, & \text{if } 1/2 < x < 1, \end{cases} \tag{7}$$

in such a way that, instead of using the variables $r_n^{(i)}$, s_n and s_{n+1} we use the binary variables $\tilde{r}_n^{(i)}$, $\tilde{s}_n$ and $\tilde{s}_{n+1}$.

Since there are only eight possible binary states, the summation in Eq. (6) has only eight terms. Besides, instead of working with the signal s_n itself, it suffices to consider the binary variables $\tilde{s}_n$ without risk of turning the decoding process unsafe. In the binary form the signal can be transmitted through any public channel. In the following we explain how the determination of the transfer entropy of each variable from the receiver network to the signal enables the receiver to decode the message.

When the receiver network computes the transfer entropy to recover the message encoded in the signal, it hardly will achieve a vanishing contribution due to the fluctuations. Moreover the receiver network sites coupled with the signal yield a transfer entropy value much higher than those uncoupled sites. A high-pass filter is used to enable the receiver network to correctly recover the message. Let m be the message sent by the emitter network and m' be the message recovered by the receiver network. The elements of the received message are given by:

$$m'^{(i)} = \Theta\left(T_{r_n^{(i)} \to s_{n+1}} - \sigma(T)\right), \tag{8}$$

where $\Theta(x)$ is the Heaviside unit step function and $\sigma(T)$ is the transfer entropy standard deviation of all network sites of the R

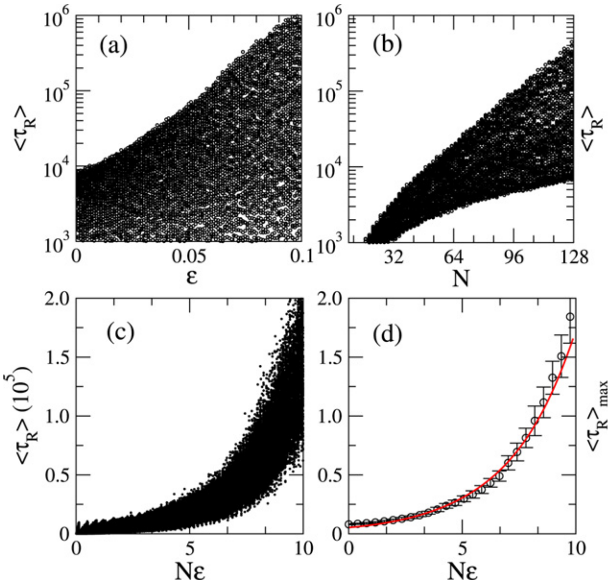

Fig. 3. Average relaxation time as a function of (a) intra-network coupling strength ε, (b) message size N, (c) $N\varepsilon$, for $N_m = 100$ different messages, each of them with $N_0 = 100$ initial conditions. (d) The upper limit for the relaxation time. The red line stands for Eq. (9). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this Letter.)

network. Hence the received message is a string of 0's and 1's if the transfer entropy of the corresponding network site is less or greater than one standard deviation of T.

The decoding process performed by the receiver network is not instantaneous, but requires a finite time interval during which a sufficiently large amount of binary signal is sent, in order to correctly decode the message. We define the relaxation time τ_R the time it takes for R to correctly decode the message, i.e. such that $m' = m$ [16]. We have observed that the value of τ_R depends on the message, for a given set of parameters. Hence we work with the average relaxation time $\langle \tau_R \rangle$, where the average is taken with respect to a number N_m of randomly chosen messages.

In numerical simulations, we consider a number $N_m = 100$ of different messages m consisting of randomly chosen strings of bits. The whole set of average relaxation times is plotted as a function of the intra-network coupling strength ε [Fig. 3(a)] and the network size N [Fig. 3(b)], where we verified that $\langle \tau_R \rangle$ grows with both parameters. So, it is suggestive to analyze the τ_R-dependency with respect to the product $N\varepsilon$ [Fig. 3(c)], which shows a growth whose upper bound is an exponential curve

$$\langle \tau_R \rangle = K e^{\kappa N \varepsilon}, \qquad (9)$$

where $K = 4.9 \times 10^3$ and $\kappa = 1/3$ were obtained by fitting the maximum relaxation time points, presented in Fig. 3(d).

During the transmission of the message it may well happen that some amount of noise corrupts the transmitted signal. It is thus important to verify if the receiver network remains able to decode correctly the message in the presence of external noise, within a time of the order of relaxation time. We consider that the signal s_n is subjected to a Gaussian noise of zero mean and variance σ. The bit error ratio (BER) is the fraction of erroneously transmitted bits with respect to the total number of bits in the message [29]. This ratio was computed as an average over $N_m = 500$ randomly chosen messages of fixed length $N = 51$. In Fig. 4 we plot the bit error ratio (BER) as a function of σ and the intra-network coupling strength ε. Note that, for $\sigma < 0.1$ BER nearly vanishes for all values of ε and, thus, there is no difference between the decoded and emitted messages for a time $n = \langle \tau_R \rangle$. Even

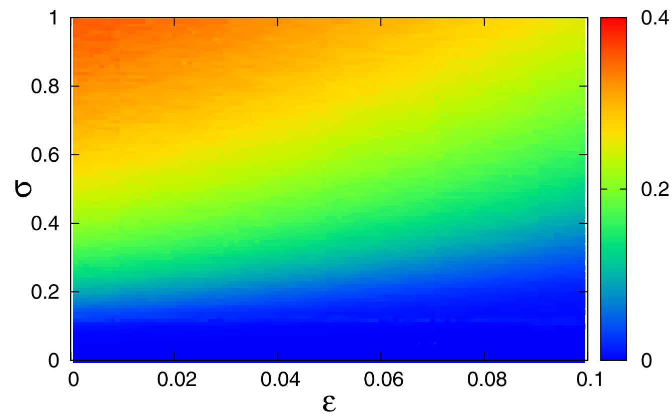

Fig. 4. Colors represent the bit error ratio as a function of the intra-network coupling strength and the noise level. The values represent an average over 500 randomly chosen messages of length 51 bits. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this Letter.)

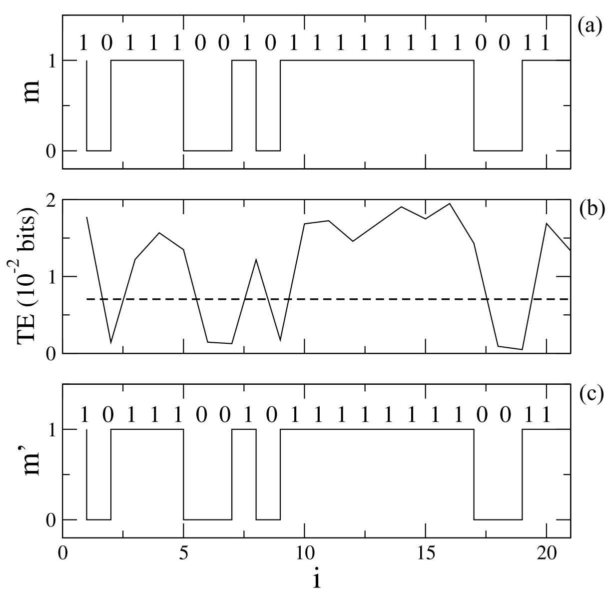

Fig. 5. (a) 21-bit message sent; (b) Transfer entropy between $\tilde{r}_n^{(i)}$ and $\tilde{s}_n$; (c) Message received.

for a stronger noise level the BER was found to be small, showing that the communication process is robust in presence of noise.

5. Application to a specific example

We now exemplify the use of the communication system described in the previous sections to transmit a specific binary message. Let us suppose that the message contains $N = 21$ bits and is given by the binary string $m = 101110010111111110011$ [Fig. 5(a)]. Hence both the emitter and receiver networks should have $N = 21$ sites. Besides having the same number of sites the networks should share a symmetric security key that is represented by the intra-network coupling strength ε, and the inter-network, from which we calculated the time to synchronize. The value of the inter-network coupling strength has been fixed as $\gamma = 1/2$. Assuming a value $\varepsilon = 0.1$, Eqs. (4) and (9) result in $\langle \tau_s \rangle = 323$ and $\langle \tau_R \rangle \approx 10^4$, respectively, for the average synchronization and relaxation times. Since these are average values, we can consider here $\tau_s = 10^3$ and $\tau_R = 2 \times 10^4$.

We couple the E and R networks following Eq. (2) and during $n = \tau_S$ iterates in the first stage of the process. At this point we expect to obtain a synchronization error $w_n < 10^{-14}$ and, if so, we truncate the state variables such that $\mathbf{e}_n = \mathbf{r}_n$ and switch off the inter-network coupling. In the beginning of the second state we use Eq. (5) to obtain a signal s_n which is transformed by Eq. (7) in a binary sequence $\tilde{s}_n$.

The receiver network keeps this binary signal during subsequent $n = \tau_R$ iterates and, through Eq. (6), computes the transfer entropy for all network sites [Fig. 5(b)]. Finally, using Eq. (8) as a high-pass filter, the receiver network decodes the message sent [Fig. 5(c)], which is clearly identical to the sent message. We remark that this process is extremely safe, since even if an eavesdropper would be able to snatch the signal that has been sent, the probability of this eavesdropper to strike all the variables $\tilde{r}_n^{(i)}$ during the time τ_R would be $P = 2^{-N\tau_R} = 2^{-63000}$. This is also the probability of striking the message sent, which is utterly insignificant.

6. Conclusions

We proposed in this work a robust and safe device for transmitting binary messages using replicas of coupled map networks. The communication system uses transfer entropy and as a consequence allows the construction of a communication system where information can only be decoded if the emitter and receiver are completely synchronous and they are exactly identical. Since nodes in each network are not synchronous, that enables the generation of a decorrelated encoded signal. This provides an extra security for the communication system, since it makes virtually impossible for an eavesdropper to discover the dynamics of the emitter.

In addition, imagine that the eavesdropper is very clever and it knows exactly the time the emitter and the receiver network take to synchronize. If it does not know exactly the connecting topology of the receiver network, it will not maintain synchronization. The eavesdropper will also not be able to verify the existence of synchronization, since all nodes in its network will be desynchronous. Suppose now that the eavesdropper knows all the secret keys (intra and inter couplings and connecting topology). Still all that is needed for the communication system to regain security is that the emitter and the receiver changes at a given time their network connecting topology, after the message has started being transmitted. This will maintain synchronization between E and R, but will make the network of the eavesdropper to become desynchronous. Both networks must have the same size N as the message itself (in number of bits).

The proposed system works in two stages: in the first one we synchronize the emitter and receiver networks. Only emitter and receiver know how much time it takes to achieve synchronization. After the first stage the inter-network coupling is switched off, but the networks remain synchronized.

In the second stage we encode a given message into a signal which is read by the receiver network using the transfer entropy between the receiver network and the signal, with a high-pass filter based on the standard deviation of the transfer entropy. The message can be decoded after a relaxation time. We obtained an expression for an upper bound of the relaxation time as a function of the intra-network coupling strength and the message size. Hence, given a message of arbitrary size the intra-network coupling strength can be chosen in order to minimize the transmission time.

The proposed device is similar to a method developed by Hung and Hu [16] but differs from it in this way: in our method we compact N bits of the message into a bit-stream of the signal, whereas in the Hung and Hu method every bit of the message is encoded in a different bit of the signal. Hence our method represents a considerable increase in the channel capacity attainable.

We considered a specific example to test this method and, comparing the message read by the receiver with the message emitted, we verified that the method is reliable. Besides this advantage, the method we propose is robust since, even in presence of external noise, the bit error ratio can be kept in sufficiently low levels, varying the intra-network coupling and noise level.

Finally, we shown that even if an eavesdropper could intercept the signal, it could not strike the message (more precisely, the probability of it is negligible). We suggest that other continuous maps as well as other intra- and inter-network couplings may also be used. Some tests with other intra-network couplings have suggested us that, for instance, the synchronization time power-law behavior remains unaltered. This was also verified when considering the logistic map for the dynamics of each network site.

Acknowledgements

This work has been made possible thanks to the partial financial support from the following Brazilian research agencies: CAPES, CNPQ and Fundação Araucária. M.S.B. acknowledges NRP.

References

[1] C. Shannon, W. Weaver, The Mathematical Theory of Communication, The University of Illinois Press, 1963.
[2] C. Shannon, Bell System Technical Journal 28 (1949) 656.
[3] T. Pecora, L.M. Caroll, Physical Review Letters 64 (1990) 821.
[4] A. Cuomo, K.M. Oppenheim, Physical Review Letters 71 (1993) 65.
[5] G. Perez, H. Cerdeira, Physical Review Letters 74 (1995) 1970.
[6] W. Kinzel, A. Englert, I. Kanter, Philosophical Transactions of The Royal Society A 368 (2010) 379.
[7] P. Stavroulakis (Ed.), Chaos Applications in Telecommunications, CRC Press, New York, USA, 2006.
[8] B. Jovic, Synchronization Techniques for Chaotic Communication Systems, Springer, Auckland, New Zealand, 2011.
[9] L. Kocarev, G. Jakimoski, T. Stojanovski, U. Parlitz, Paper presented at the Conference on the International Symposium on Circuits and System, 1998.
[10] M. da Silva Baptista, Physics Letters A 240 (1998) 50.
[11] G. Alvarez, S. Li, International Journal of Bifurcation and Chaos 16 (2006) 2129.
[12] E. Alvarez, A. Fernandez, J. Garcia, P. Jimenez, A. Marcano, Physics Letters A 263 (1999) 373.
[13] S. Li, G. Chen, K.E.A. Wong, Physics Letters A 332 (2004) 368.
[14] L. Kocarev, IEEE Circuits and Systems 1 (2002) 6.
[15] M. Ding, E. Ott, Physical Review E 49 (1994) R945.
[16] C. Hung, Y.C. Hu, Physical Review Letters 101 (2008) 244102(4).
[17] T. Schreiber, Physical Review Letters 85 (2000) 4.
[18] G. Vidal, M.S. Baptista, H. Mancini, International Journal of Bifurcation and Chaos 22 (10) (2012), http://dx.doi.org/10.1142/S0218127412502434.
[19] H. Jaeger, Technical Report GMD, German National Research Center for Information Technology, Report 148, 2001.
[20] L. Appeltant, M. Soriano, G. Sande, Nature Communications 2 (2010) 468.
[21] J. Crutchfield, K. Kaneko, Phenomenology of Spatio-Temporal Chaos, vol. 1, World Scientific, 1987.
[22] D. Vasconcelos, R. Viana, S. Lopes, A. Batista, S. Pinto, Physica A—Statistical Mechanics and its Applications 343 (2004) 201.
[23] R.F. Pereira, S.E. de S. Pinto, S.R. Lopes, Physica A—Statistical Mechanics and its Applications 389 (2010) 5279.
[24] V. Ahlers, A. Pikovsky, Physical Review Letters 88 (2002) 254101(4).
[25] M. Cencini, C. Tessone, A. Torcini, Chaos 18 (2008) 037125.
[26] I. Szendro, M. Rodrigues, J. López, Europhysics Letters 86 (2009) 2008.
[27] E. Ott, Chaos in Dynamical Systems, Cambridge University Press, 1993.
[28] G.S. Vernam, Journal of the IEEE 55 (1926) 109.
[29] P. Stavroulakis (Ed.), Elements of Information Theory, John Wiley and Sons, New Jersey, USA, 2006.