Spying on chaos-based cryptosystems with reservoir computing

Piotr Antonik, Marvyn Gulina, Jaël Pauwels, Damien Rontani, Marc Haelterman, Serge Massar

Spying on chaos-based cryptosystems with reservoir computing

Piotr Antonik
LMOPS EA-4423 Lab
CentraleSupélec, Université Paris-Saclay
Metz, France
piotr.antonik@centralesupelec.fr

Marvyn Gulina
Centre Namurois des Systèmes Complexes
Université de Namur
Namur, Belgium

Jaël Pauwels
Laboratoire d'Information Quantique
Université libre de Bruxelles
Brussels, Belgium

Damien Rontani
Chair in Photonics
CentraleSupélec, Université Paris-Saclay
Metz, France

Marc Haelterman
Service OPERA-Photonique
Université libre de Bruxelles
Brussels, Belgium

Serge Massar
Laboratoire d'Information Quantique
Université libre de Bruxelles
Brussels, Belgium

***Abstract*—Reservoir computing is a machine learning approach to designing artificial neural networks. Despite the significant simplification of the training process, the performance of such systems is comparable to other digital algorithms on a series of benchmark tasks. Recent investigations have demonstrated the possibility of performing long-horizon predictions of chaotic systems using reservoir computing. In this work we show that a trained reservoir computer can reproduce sufficiently well the properties a chaotic system, hence allowing full synchronisation. We illustrate this behaviour on the Mackey-Glass and Lorenz systems. Furthermore, we show that a reservoir computer can be used to crack chaos-based cryptographic protocols and illustrate this on two encryption schemes.**

***Index Terms*—reservoir computing, chaos synchronisation, chaos-based cryptography, eavesdropping**

I. Introduction

Is it possible to emulate a non-linear chaotic dynamical system with a fundamentally different non-linear dynamical system? This question has been answered positively in the context of a machine learning technique known as reservoir computing [1]–[4].

Reservoir Computing (RC) is a set of methods for designing and training artificial neural networks, introduced independently in [5] and [6]. The underlying idea of these techniques is that one can exploit the dynamics of a recurrent nonlinear network to process time series without training the network itself, but simply adding a general linear readout layer and only training the latter. This results in a system that is significantly easier to train (the learning is reduced to solving a system of linear equations [7]), yet powerful enough to match other algorithms on a series of benchmark tasks. For instance, RC has been successfully applied to channel equalisation [1], phoneme recognition [8], and won an international competition on prediction of future evolution of financial time series [9].

In the case of emulating a dynamical system, the reservoir is first driven by the state of the given system, and trained to predict this state one time step in the future. After training, the output of the reservoir is fed back into itself, whereupon it will develop autonomous dynamics that are – one hopes – close to those of the original dynamical system.

Reservoir computing was originally used for forecasting the trajectories of chaotic dynamical systems, where it reached record forecasting horizons [1] (see also the recent improvements in [10]). This method was also implemented experimentally in [2], where it was shown that the reservoir emulator had similar dynamics to the original system in terms of RF spectrum, Lyapunov exponents, and "randomness" properties. It was also used in [3] to infer the values of hidden degrees of freedom of the dynamical system, and in [4] to estimate its Lyapunov exponents. These works suggest that a trained reservoir computer can emulate another, a priori completely different, even chaotic, dynamical system.

In the first part of this work, we demonstrate that a trained reservoir computer captures a large part of the characteristics of the dynamics of the original system. That is, if weakly driven by the original system, the reservoir computer will synchronise with it. We illustrate this phenomenon on two examples, the Lorenz [11] and Mackey-Glass systems [12]. The phenomenon of synchronisation is one of the most surprising aspects of chaos theory, and has been extensively studied, see e.g. the review [13]. However, our results appear in great contrast with what was known about synchronisation of chaotic systems, in the sense that two twin physical systems were required to achieve similar properties of the generated chaotic time series.

After the discovery of chaos synchronisation, considerable effort was devoted to trying to use this effect and the unpredictability of chaotic systems to hide secret messages. Communication protocols based on chaos synchronisation [14]

This work was supported by the Interuniversity Attraction Poles Program (Belgian Science Policy) Project Photonics@be IAP P7-35, by the Fonds de la Recherche Scientifique (FRS-FNRS), and by the Action de Recherche Concertée of the Fédération Universitaire Wallonie-Bruxelles through Grant No. AUWB-2012-12/17-ULB9. P.A. and D.R gratefully acknowledge the support of AFOSR (grants No. FA-9550-15-1-0279 and FA-9550-17-1-0072) and Région Grand-Est.

were proposed in the early 1990s [15], [16]. In this type of systems, a message is embedded within a chaotic carrier in the emitter, and recovered after transmission by a receiver upon synchronisation with the emitter. Chaotic communication systems are particularly attractive due to the broadband power spectrum of the generated waveforms, high rates of information transmission, and sufficient efficiency at relatively low signal-to-noise ratios. Besides, many chaotic communication schemes are simply realised and demonstrate a rich variety of different oscillating regimes [17]. Optical systems provide simple ways of generating high-dimensional chaotic carriers, offering a substantial security level and the possibility of very high transmission rates [18]. Early experiments demonstrated successful back-to-back communications in all-optical [19] and opto-electronic [20] systems with high bit rates (> 1 Gb/s). Recent work [21] reports a successful long-distance field experiment.

The security of chaos-based transmissions relies on the fact that the emitting and receiving parties must have similar copies of a chaotic attractor, that is very challenging to manufacture for a third party, without any knowledge of its internal structure and parameters. However, a potential eavesdropper could crack the chaotic masking with a device capable of emulating a chaotic system, such as the reservoir computer.

In the second part of this work, as an application of our results on chaos synchronisation, we consider using the reservoir computer to crack two chaos-based encryption schemes with Mackey-Glass and Lorenz chaotic carriers. The successful results we obtain suggest that hardware chaos-based cryptosystems could be cracked by hardware reservoir computers, as these have been implemented physically with good performance and high speed, see [22] for a review.

II. Reservoir computing

The reservoir computer used in this work is an echo state network, introduced in [1], [5]. The reservoir states vector x, consisting of N neurons, is updated following the equation

$$x(n) = (1 - Ca)x(n-1) + C\tanh\left(w_{\text{in}}u + Wx(n-1) + w_{\text{back}}d(n-1)\right), \quad (1)$$

where $n \in \mathbb{Z}$ is the discrete time, C is a timescale constant, a is the leak rate, W is a $N \times N$ matrix of internal connection weights, w_{back} is N-size weight vector for for feedback connections from the output to the reservoir and w_{in} is a N-size vector and u is a constant.

The elements of w_{in}, W and w_{back} are chosen from a uniform distribution over the interval $[-1, +1]$. A reservoir computer must be not too far from the edge of chaos to exhibit good performance. To this end the matrix W is then rescaled to adapt its spectral radius. The vectors w_{in} and w_{back} are possibly also rescaled to adapt the strength of the input and feedback. Throughout this work the input bias is fixed to $u = 0.2$.

The output equation of a single-output network is given by a dot product

$$y(n) = w_{\text{out}} \cdot x(n), \quad (2)$$

where w_{out} are $N+1$ output weights (also known as the output mask).

The reservoir computer is operated in two stages: a training phase and a free run. During the training phase, the system is driven by a chaotic time series, denoted by $s(n)$. The evolution of the reservoir during training is given by Eq. 1, supplemented by

$$d(n) = s(n) \qquad \text{(during training)}. \quad (3)$$

The training phase is used to optimise the readout weights w_{out} so that the reservoir predicts the next point $s(n)$ in the input chaotic time series, given the previous points $s(n-1), s(n-2), \ldots$. To this end we minimise the Normalised Mean Square Error (NMSE), given by

$$\text{NMSE} = \frac{\left\langle \left(y(n) - s(n)\right)^2 \right\rangle}{\left\langle \left(s(n) - \langle s(n) \rangle\right)^2 \right\rangle}. \quad (4)$$

Minimising the NMSE with respect to the readout weights gives rise to a system of linear equations that are readily solved with e.g. a simple linear regression.

After the training, the readout weights w_{out} are fixed and the teacher signal $d(n)$ is replaced by the output signal $y(n)$, so that the reservoir becomes autonomous. The evolution of the reservoir computer during the autonomous run is given by Eqs. 1 and 2, supplemented by

$$d(n) = y(n) \qquad \text{(during autonomous run)}. \quad (5)$$

III. Training on the Mackey-Glass and Lorenz systems

For illustrative purposes in this work, we use the one-dimensional Mackey-Glass (MG) delay equation and the tri-dimensional Lorenz system.

The Mackey-Glass delay differential equation

$$\frac{dx}{dt} = \beta \frac{x(t-\tau)}{1 + x^n(t-\tau)} - \gamma x \quad (6)$$

with τ, γ, β, $n > 0$ was introduced to illustrate the appearance of complex dynamics in physiological control systems [12]. To obtain chaotic dynamics, we set the parameters as in [1]: $\beta = 0.2$, $\gamma = 0.1$, $\tau = 17$ and $n = 10$. With these settings, the highest Lyapunov exponent is $\lambda = 0.006$ [1].

Eq. 6 was integrated using Matlab's `dde23` solver with the initial condition $x(0) = 0.5$ and integration step of 0.5 for 7000 timesteps. The first 1000 transient values were discarded and the remaining data was split into 3000 training and 3000 test inputs.

For the MG task, we used a reservoir with $N = 1500$ neurons, the matrix W was rescaled to a spectral radius of 0.79, while the vectors w_{in}, w_{back} were not rescaled, and we set $C = 1$, $a = 0.9$. These heuristic parameters were found to provide good results.

At the training stage we obtained an error of NMSE $= 3 \times 10^{-9}$. During the free run, the error gradually increases, as the reservoir output signal slowly deviates from the target trajectory on the Mackey-Glass attractor. Nevertheless, the

system manages to generate the desired output for several hundreds of time steps with reasonable precision.

The Lorenz equations, a system of three ordinary differential equations

$$\frac{dx}{dt} = \sigma\left(y - x\right), \tag{7a}$$
$$\frac{dy}{dt} = -xz + rx - y, \tag{7b}$$
$$\frac{dz}{dt} = xy - bz, \tag{7c}$$

with $\sigma, r, b > 0$, was introduced as a simple model for atmospheric convection [11]. The system exhibits chaotic behaviour for $\sigma = 10, b = 8/3$ and $r = 28$ [23], that we used in this study. This yields a chaotic attractor with the highest Lyapunov exponent of $\lambda = 0.906$ [1].

The Lorenz Eqs. 7 were integrated using Matlab's `ode45` routine with an integration step of 0.02 for 10000 timesteps. We only used the x coordinate of the chaotic system, which was rescaled by the factor of 0.01, as in [1]. The first 1000 transient values were discarded and the remaining data was split into 6000 training and 3000 test inputs.

For the Lorenz task, we used a reservoir of size $N = 1500$. We set the spectral radius of the weight matrix W to 0.97, the input and feedback weights w_{in} and w_{back} were rescaled to belong to the interval $[-0.5, 0.5]$, and we set $C = 0.44$ and $a = 0.9$. These heuristic parameters were found to provide good results.

We obtained a training error of NMSE $= 3\times10^{-8}$. The error is one order of magnitude higher here than in the Mackey-Glass case because the Lorenz systems is more complex and more chaotic.

IV. Synchronising a reservoir computer on the Mackey-Glass and Lorenz systems

Let $s(n)$ be the time series of the chaotic system with which one wishes to synchronise. We first train the reservoir to predict the next time step in the series, as described above. Next we start an autonomous run in which the reservoir follows its own dynamics, given by Eqs. 1, 2, 5. At time $n = n_0$, we start weakly driving the reservoir with the chaotic time series $s(n)$. That is, its dynamics is given by Eqs. 1, 2, supplemented by

$$d(n) = (1 - q)y(n) + qs(n) \qquad \text{(when locked)}. \tag{8}$$

with $0 \leq q \leq 1$.

Figures 1 and 2 illustrate how the trained reservoir can lock onto the MG and Lorenz systems. It should be noted that during the synchronisation phase, the NMSE stays at a relatively high value, whereas if we were to synchronise two identical MG or Lorenz systems, the NMSE would decrease until it reached the machine precision. This is because the trained reservoir does not reproduce exactly the dynamics of the MG or Lorenz system.

V. Cracking chaos-based cryptography

Chaos-based cryptography systems are based on Alice and Bob having two identical (or nearly identical) chaotic systems, and using the unpredictable nature of the chaotic system to mask the message Alice wants to transmit to Bob. Many different methods have been proposed to mask the message, see e.g. [24]. The cryptanalysis problem is for an eavesdropper (Eve) who has access to the public (encrypted) message sent by Alice to recover the secret (plain) message.

There have been many attempts to crack such cryptosystems, see e.g. [25]. These approaches are based on fitting the unknown parameters in Alice's chaotic device, thereby enabling Eve to reproduce Bob's decoder, and hence recover the message.

The fact that a reservoir computer can be trained to emulate chaotic systems, to the extent that the trained reservoir will synchronise with the original chaotic system (as demonstrated in Sec. IV) suggests that reservoir computing could form the basis for an alternative, conceptually different, approach to cracking chaos-based cryptography.

We illustrate this novel approach on two examples of encryption schemes.

A. Superposition scheme

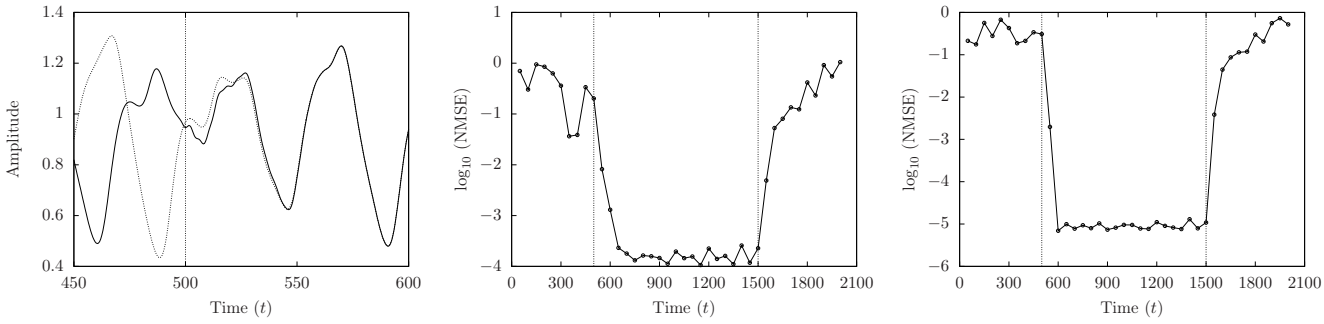
Let $m(t)$ be the transmitted message, and $a(t)$ and $b(t)$ the outputs of Alice's and Bob's chaotic systems, respectively. The idea behind this scheme is to create an encrypted signal

$$s(t) = a(t) + m(t), \tag{9}$$

with $|m(t)| \ll |a(t)|, \forall t$. Subtracting the chaotic carrier, Bob obtains the message $\tilde{m}(t) = s(t) - b(t)$. After synchronisation of Alice's and Bob's chaotic systems, so that $a(t) \simeq b(t)$, Bob ends up with $\tilde{m}(t) \simeq m(t)$.

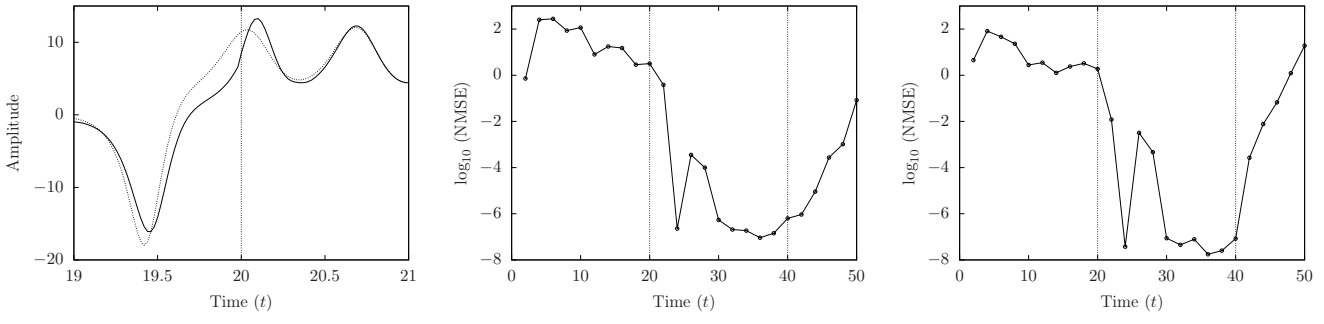
In this work, the message sent by Alice to Bob is composed of a sequence of 4000 random bits. The message is preceded by a series of 400 null bits (that is, no message is added to the chaotic carrier), so that to transmit an example of the chaotic carrier signal in order to allow Bob's system to lock on to the Alice's one. That is, the synchronisation is only performed during a relatively short time interval.

As the synchronisation between two chaotic systems is sensitive to noise, the superposition scheme, that fully relies on synchronisation, is also affected by noise. Figure 3(a) illustrates the message decrypted by Bob. The transmitted bits are shown with circles, and form two thick horizontal lines, as we display a long part of the message. The decrypted message is displayed with crosses, connected with a dashed line. We set a high level of transmission noise ($A_\nu = 10^{-5}$) to make the desynchronisation effect more apparent. The first decrypted bits match the encoded ones. As we advance further through the message, the decrypted signal oscillates around the correct values, before these oscillations grow bigger in amplitude than the message itself, thus compromising the decryption. Starting from $t = 1200$, the decrypted signal looks more and more similar to the Mackey-Glass chaotic carrier, which means that Bob's system has lost synchronisation with Alice's generator.



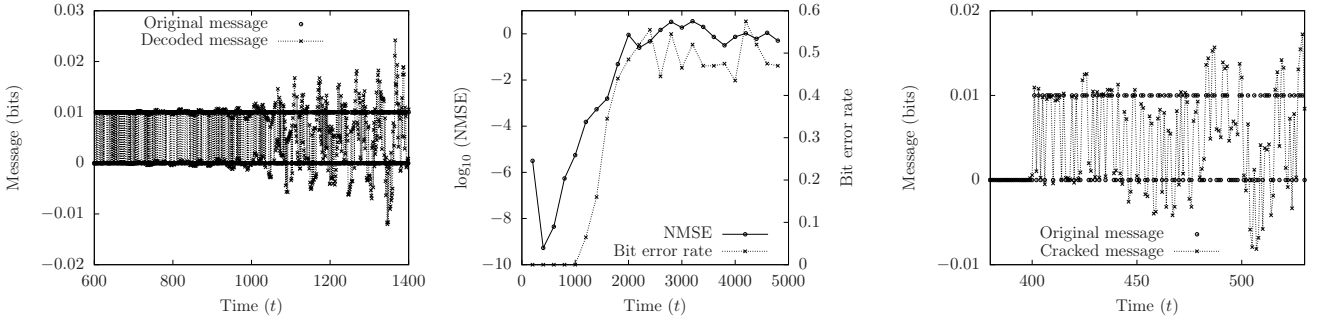
(a) (b) (c)

Fig. 1. Synchronisation of a trained reservoir computer on the Mackey-Glass system, integrated from $x_{\mathrm{MG}}(0) = 0.5$. The reservoir, started from a different initial condition $x_{\mathrm{RC}}(0) = 0.9$, is autonomous until $t = 500$, whereupon we set the driving strength to $q = 0.25$. Plot (a) only depicts the region of interest around $t = 500$, where the reservoir (plotted with a solid line) synchronises with the chaotic system (traced with a dotted line). Plot (b) shows the evolution of the NMSE, averaged over 100-timestep intervals, for the entire duration of the simulation, where the coupling is removed ($q = 0$) at $t = 1500$. Plot (c) illustrates the same scenario with a higher coupling strength $q = 0.5$. The synchronisation is quicker in this case, as can be seen from the steeper slope, and the resulting NMSE is lower. These observations suggest that the synchronisation phenomenon is more efficient with higher coupling ratios, and may disappear for q below a certain threshold value.



(a) (b) (c)

Fig. 2. Synchronisation of a trained reservoir computer on the Lorenz system, integrated from $(x_0, y_0, z_0)_{\mathrm{LZ}} = (10, 0, 0)$. The reservoir, integrated from $(x_0, y_0, z_0)_{\mathrm{RC}} = (4, 0, 0)$ is autonomous until $t = 20$, whereupon we set the driving strength to $q = 0.25$. Plot (a) only depicts the region of interest around $t = 20$, where the reservoir (plotted with a solid line) synchronises with the chaotic system (traced with a dotted line). Plot (b) shows the evolution of the NMSE, averaged over 100-timestep intervals, for the entire duration of the simulation, where the coupling is removed ($q = 0$) at $t = 40$. Plot (c) illustrates the same scenario with a higher coupling strength $q = 0.5$. Here again, the synchronisation is slightly quicker, and the resulting NMSE is slightly lower.



(a) (b) (c)

Fig. 3. Decryption of the superposition scheme with the Mackey-Glass carrier. (a) Illustration of Bob's decryption process in presence of high transmission noise $A_\nu = 10^{-5}$. The decrypted message starts at $t = 400$, as explained above, but we focus here on a specific region to demonstrate the desynchronisation issues. The encoded bits form the two horizontal lines because a long portion of the message is displayed. The decrypted bits, shown with crosses, first match the encoded ones. From approximately $t = 800$, the decrypted signal starts oscillating and after $t = 1200$ the decrypted bits can no longer be identified. This shows that Bob's chaotic system is no longer locked on to the one of Alice. (b) Comparison of the synchronisation error (solid line, averaged over intervals of 200 timesteps) between Alice's and Bob's carriers, and the bit error rate (dashed line) of the decrypted message. As the former crosses the NMSE $= 10^{-4}$ mark, the bit error start growing as well, until it reaches the 0.5 value, indicating randomly guessed bits. The initial dip of the NMSE corresponds to the locking of Bob's systems on Alice's. (c) Eve's attempt to crack the encoded message, shown with circles. The null bits before $t = 400$ correspond to the synchronisation period, as explained above. Eve's cracked bits are plotted with crosses. She managed to recover roughly 80 first bits of the message. After that, her signal starts oscillating in the manner of the chaotic carrier, because the reservoir computer lost synchronisation with Alice's system.

Figure 3(b) illustrate the same case from the synchronisation point of view between the chaotic carriers of Alice and Bob. The solid line corresponds to the synchronisation error, that steadily grows until NMSE $\sim 10^{-1}$, where both systems are completely out of sync. The dashed line displays the bit error rate on the decoded message, which grows together with the synchronisation error before reaching 0.5, where the decoded bits are guessed at random.

Figure 3(c) shows the message decrypted by Eve. In this example of a noiseless transmission, Eve manages to accurately decrypt the first 80 bits of the message, before bit-errors start to appear in the cracked message. After roughly 100 timesteps, the reservoir computer looses synchronisation with the chaotic carrier to the point where the recovered bits become useless.

These results show that Eve finds herself in the same situation as Bob in case of a noisy communication, discussed above. Since the reservoir computer can only emulate the chaotic system with a limited precision, it desynchronises fairly quickly from the Alice's system, thus making further decryption impossible. Nevertheless, this simple proof-of-principle experiment demonstrates that Eve is capable of cracking some part of the message without any information about the chaotic carrier used for transmission.

To check the influence of the chaotic carrier signal, we now switch from the Mackey-Glass system to Lorenz. Again, we start with Bob in possession of a valid copy of the Lorenz system with identical parameters to Alice. The communication scheme is the same as above, with one exception: 1600 null bits are sent for the synchronisation process, as Lorenz dynamics seems more complex to capture than Mackey-Glass. Therefore, since the system is integrated with a step of $h = 0.02$, the message starts at $t = 32$.

Figure 4(a) illustrates the scenario with a high noise level $A_\nu = 10^{-5}$. Since the Lorenz system is more chaotic, the desynchronisation effect is quicker and more apparent in this case, compared to the Mackey-Glass carrier. The encrypted bits are shown with circles. As we display a large portion of the message, the bits fuse into two horizontal lines. The graph shows how the decrypted bits, displayed with crosses, match the encoded ones. While the beginning of the message is recovered accurately, at approximately $t = 36$ the decoded signal starts oscillating, before becoming meaningless at $t = 40$. Again, this shows that Bob's system is no longer locked onto the chaotic carrier.

Figure 4(b) displays the carrier NMSE and the bit error rate for the same case. Similarly to the situation with the Mackey-Glass carrier (see Fig. 3(b)), the growth of the synchronisation error is followed by a growth of the bit error rate. The only difference with the Lorenz carrier is that, since it is more chaotic, the desynchronisation happens faster.

Figure 4(c) shows the message decrypted by Eve. In this example of a noiseless transmission, Eve manages to accurately decrypt the first 30 bits of the message, before bit-errors start to appear in the cracked message. After roughly 50 timesteps, the reservoir computer looses synchronisation with the chaotic carrier to the point where the recovered bits become useless.

These results show that, again, Eve finds herself in the same situation as Bob in case of a noisy communication, discussed above – the reservoir computer desynchronises fairly quickly from the Alice's system because of the limited emulation precision, thus making further decryption impossible. Nevertheless, Eve could recover some part of the message without any knowledge of the chaotic carrier.

B. Nonlinear mixing scheme

Here we study an encryption scheme introduced in [17] (specifically, the III/1 scheme of this paper).

To encode her message, Alice uses a delay dynamical system in which she injects her message $m(t)$. Her dynamical system obeys the equation

$$\epsilon \dot{x}(t) = -\gamma x(t) + f\left[x(t-\tau) + m(t-\tau)\right], \qquad (10)$$

where τ is the delay. Alice sends $s(t) = x(t) + m(t)$ to Bob, i.e. the argument of $f[.]$ in Eq. 10.

To decrypt the message, Bob uses the same delay system, but in an open loop configuration to obtain the variable $x'(t)$ given by

$$\epsilon \dot{x'}(t) = -\gamma x'(t) + f\left[s(t-\tau)\right]. \qquad (11)$$

To recover the message, Bob then simply computes the difference $m'(t) = s(t) - x'(t)$.

In order to crack this system, we suppose that Eve has access to a plain text attack, i.e. she has access to both $s(t)$ and $m(t)$ during some time interval. She then trains a reservoir computer which receives as input $s(t)$ to compute $x(t+\tau)$, i.e. to reproduce the operation of Bob's decoder Eq. 11. Once the training is accomplished, she can use her reservoir computer to replace Bob's system, thereby recovering encrypted messages.

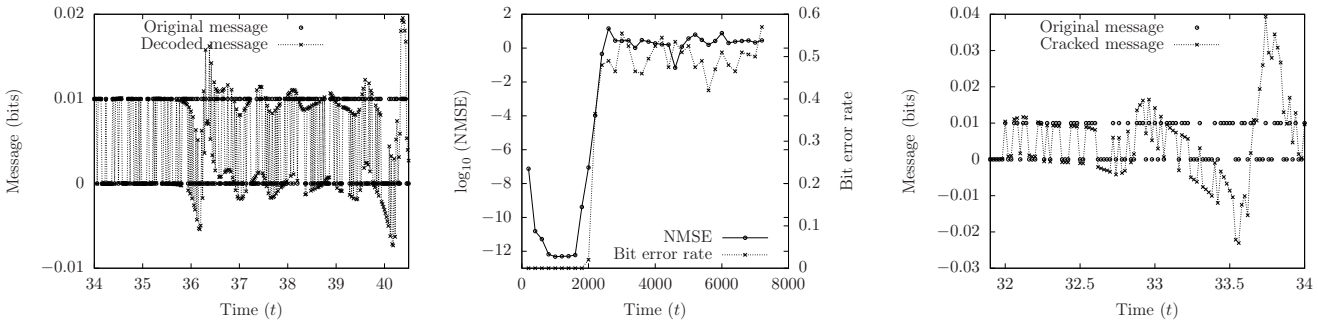
To illustrate this, we used the MG system Eq. 6 with the same parameters used elsewhere in this work ($\beta = 0.2$, $\gamma = 0.1$, $\tau = 17$ and $n = 10$). We use a discretised time step of 0.02. As message we consider a frequency-modulated harmonic signal of the form

$$m(t) = A \sin\left[2\pi f_c t - B\cos(2\pi f_m t)\right], \qquad (12)$$

where $f_c = 5 \times 10^{-3}$ is the central frequency of the power spectrum of the signal, $B = 3$ is the frequency modulation index, $f_m = 5 \times 10^{-5}$ is the modulation frequency and $A = 0.01$ is the amplitude of the message, chosen to ensure that the information signal comprises 1% of the amplitude of the chaotic carrier. The message and the values of the parameters are identical to those used in [17].

To crack the system, Eve used a reservoir computer with $N = 1500$ internal nodes and trained on a message comprising 12000 time steps. The spectral radius of the weight matrix W was set to 0.97, the input and feedback weights w_{in} and w_{back} were rescaled with global coefficients of 0.9 and 0.8, respectively, and we set $C = 0.44$ and $a = 0.9$. We obtained a training error of NMSE $= 5.8 \times 10^{-6}$.

Using this trained RC, Eve can now try to recover an unknown message sent by Alice. The results are presented in Fig. 5, where we compare decryption by Bob and Eve.



(a) (b) (c)

Fig. 4. Decryption of the superposition scheme with the Lorenz carrier. (a) Illustration of Bob's decryption process in presence of high transmission noise $A_\nu = 10^{-5}$. The decrypted message starts at $t = 32$, as explained above, but we focus here on a specific region to demonstrate the desynchronisation issues. The encoded bits form the two horizontal lines because a long portion of the message is displayed. The decrypted bits, shown with crosses, first match the encoded ones. From approximately $t = 36$, the decrypted signal starts oscillating and after $t = 40$ the decrypted bits can no longer be identified. This shows that Bob's chaotic system is no longer locked on to the one of Alice. (b) Comparison of the synchronisation error (solid line, averaged over intervals of 200 timesteps) between Alice's and Bob's carriers, and the bit error rate (dashed line) of the decrypted message. Because of the high Lyapunov exponent of the Lorenz carrier, the desynchronisation process is very fast here, and so is the growth of the bit error rate. The initial dip of the NMSE corresponds to the locking of Bob's systems on Alice's. (c) Eve's attempt to crack the encoded message, shown with circles. The null bits before $t = 32$ correspond to the synchronisation period, as explained above. Eve's cracked bits are plotted with crosses. She managed to recover roughly 30 first bits of the message. After that, her signal starts oscillating in the manner of the chaotic carrier, because the reservoir computer lost synchronisation with Alice's system.

We depict the raw messages decoded by Bob and Eve, their spectra, and the messages after processing with a low-pass filter. We see that both Bob's and Eve's decoded messages are corrupted by high frequency noise, with Eve suffering from much higher levels of noise.

VI. Conclusion

In this work we addressed the question of the quality of emulation of a nonlinear dynamical system by a reservoir computer. We have shown that a trained RC captures a large part of the characteristics of the primary system, to the point where synchronisation is possible. We demonstrated this phenomenon by synchronising a weakly driven reservoir computer onto Mackey-Glass and Lorenz chaotic systems. Furthermore, we tackled two encryption schemes in chaos-based cryptography and managed to successfully decode the secret messages in both cases. This study thus asserts the capacity of a reservoir computer to faithfully emulate dynamical systems, even in a chaotic regime, and opens new applications in the field of cryptography.

Acknowledgment

The authors thank Guy Van Der Sande and Guy Verschaffelt for insightful discussions.

References

[1] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, pp. 78–80, 2004.

[2] P. Antonik, M. Haelterman, and S. Massar, "Brain-inspired photonic signal processor for generating periodic patterns and emulating chaotic systems," *Phys. Rev. Applied*, vol. 7, p. 054014, May 2017.

[3] Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, "Reservoir observers: Model-free inference of unmeasured variables in chaotic systems," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 4, p. 041102, 2017.

[4] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, "Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, no. 12, p. 121102, 2017.

[5] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks - with an Erratum note," *GMD Report*, vol. 148, 2001.

[6] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural comput.*, vol. 14, pp. 2531–2560, 2002.

[7] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comp. Sci. Rev.*, vol. 3, pp. 127–149, 2009.

[8] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, "Phoneme recognition with large hierarchical reservoirs," *Adv. Neural Inf. Process. Syst.*, vol. 23, pp. 2307–2315, 2010.

[9] "The 2006/07 forecasting competition for neural networks & computational intelligence." http://www.neural-forecasting-competition.com/NN3/, 2006.

[10] L. Grigoryeva and J.-P. Ortega, "Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems," *arXiv preprint arXiv:1712.00754*, 2017.

[11] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[12] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[13] S. Boccaletti, J. Kurths, G. Osipov, D. Valladares, and C. Zhou, "The synchronization of chaotic systems," *Physics reports*, vol. 366, no. 1, pp. 1–101, 2002.

[14] P. Ashwin, "Nonlinear dynamics: Synchronization from chaos," *Nature*, vol. 422, no. 6930, pp. 384–385, 2003.

[15] K. M. Cuomo, A. V. Oppenheim, and S. H. Strogatz, "Synchronization of lorenz-based chaotic circuits with applications to communications," *IEEE Transactions on circuits and systems II: Analog and digital signal processing*, vol. 40, no. 10, pp. 626–633, 1993.

[16] P. Colet and R. Roy, "Digital communication with synchronized chaotic lasers," *Optics letters*, vol. 19, no. 24, pp. 2056–2058, 1994.

[17] M. Prokhorov and V. Ponomarenko, "Encryption and decryption of information in chaotic communication systems governed by delay-differential equations," *Chaos, Solitons & Fractals*, vol. 35, no. 5, pp. 871–877, 2008.

[18] C. R. Mirasso, P. Colet, and P. García-Fernández, "Synchronization of chaotic semiconductor lasers: Application to encoded communications," *IEEE Photonics Technology Letters*, vol. 8, no. 2, pp. 299–301, 1996.

[19] G. D. Vanwiggeren and R. Roy, "Communication with chaotic lasers," *Science*, vol. 279, no. 5354, pp. 1198–1200, 1998.

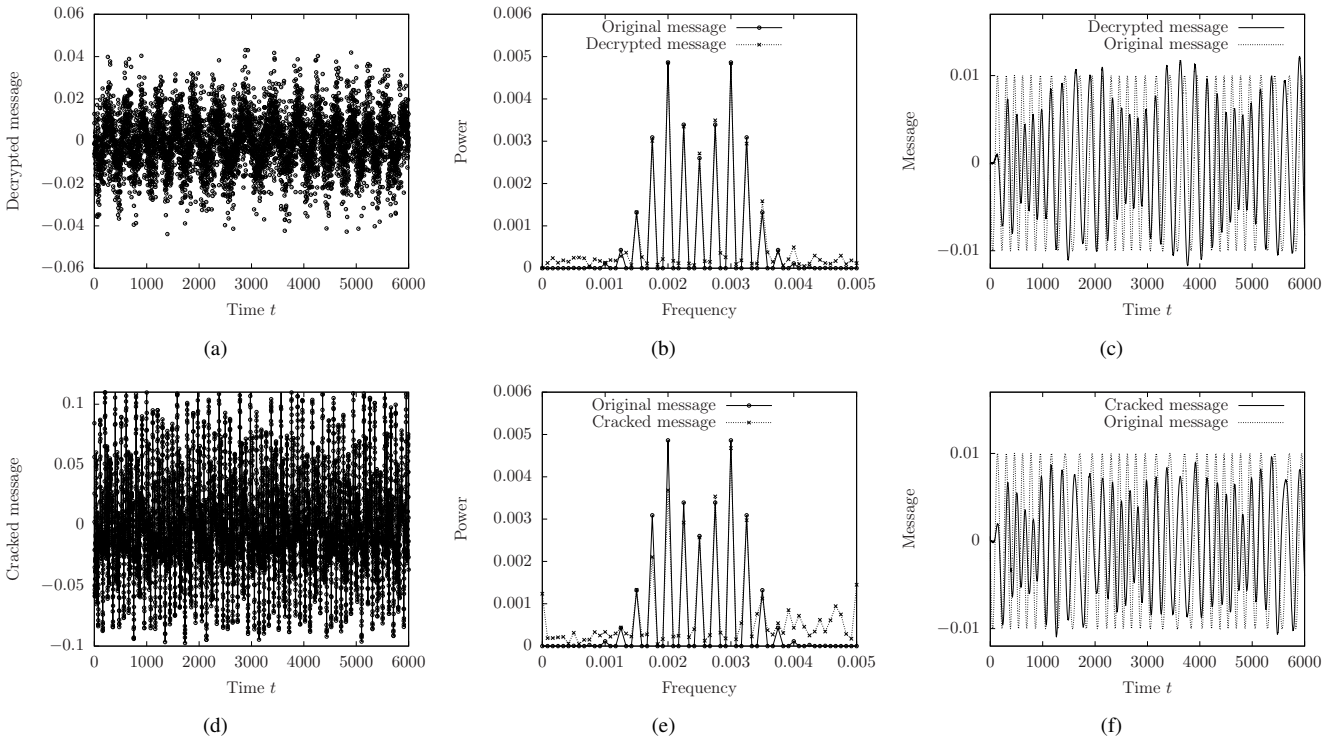


Fig. 5. Comparison of messages recovered by Bob (panels (a), (b) and (c)) and Eve (panels (d), (e) and (f)). The panels (a) and (d) show the raw outputs of Bob's decoder and Eve's reservoir computer. While the frequency modulation message can be recognised in panel (a), not much can be taken from Eve's data in panel (d). Panels (b) and (e) displays frequency spectra of the raw messages. The plots show that both Bob and Eve have recovered correct spectra of the original message, but with significant amount of high-frequency noise (especially in the case of Eve). Panels (c) and (f) present the recovered messages processed by a low-pass filter. Both Bob and Eve could accurately recover the frequency modulation, although with some insignificant amplitude variations.

[20] J.-P. Goedgebuer, L. Larger, and H. Porte, "Optical cryptosystem based on synchronization of hyperchaos generated by a delayed feedback tunable laser diode," *Physical Review Letters*, vol. 80, no. 10, p. 2249, 1998.

[21] A. Argyris, D. Syvridis, L. Larger, V. Annovazzi-Lodi, *et al.*, "Chaos-based communications at high bit rates using commercial fibre-optic links," *Nature*, vol. 438, no. 7066, p. 343, 2005.

[22] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.

[23] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential equations, dynamical systems, and an introduction to chaos*. Academic press, Boston, MA, 2003.

[24] L. Kocarev, "Chaos-based cryptography: a brief overview," *IEEE Circuits and Systems Magazine*, vol. 1, no. 3, pp. 6–21, 2001.

[25] F. Anstett, G. Millerioux, and G. Bloch, "Chaotic cryptosystems: Cryptanalysis and identifiability," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 12, pp. 2673–2680, 2006.