

Plain convolution encryption as an alternative to overcoming the limitations of synchronization-based methods

F. Rosales-Infante^{a,*}, M. L. Romero-Amezcu^{b,*}, I. Álvarez-Rios^{a,*}, and F. S. Guzmán^{a,f}

^a*Instituto de Física y Matemáticas, Universidad Michoacana de San Nicolás de Hidalgo, Edificio C-3, Cd. Universitaria, 58040 Morelia, Michoacán, México.*

*e-mails: *flavio.rosales.infante@umich.mx; *ivan.alvarez@umich.mx; †francisco.s.guzman@umich.mx*

^b*Facultad de Ciencias Físico Matemáticas, Universidad Michoacana de San Nicolás de Hidalgo, Edificio ALFA, Cd. Universitaria, 58040 Morelia, Michoacán, México.*

*e-mails: *maria.romero.amezcua@umich.mx*

Received 8 January 2025; accepted 28 February 2025

This paper revisits the send/retrieve message process using synchronization of the Lorenz system with a monochromatic message. We analyze how the fidelity of the retrieved signal depends on the message frequency and demonstrate message hacking through Fourier spectrum analysis. Various parameters affecting fidelity and noise in the hacked signal are also examined. Additionally, we transmit text messages recovered through synchronization and investigate their vulnerability to hacking. As a countermeasure, we propose a method to send both types of messages using the convolution as the encryption function to hide the message in the chaotic signal. This approach enhances retrieval fidelity and significantly increases resistance to hacking compared to synchronization-based methods.

Keywords: Lorenz system; Fourier spectrum analysis.

DOI: <https://doi.org/10.31349/RevMexFis.71.040602>

1. Introduction

The application of chaos theory to secure communication has led to a variety of encryption methods that take advantage of the unpredictability of chaotic systems. Early developments using chaos synchronization, demonstrated the potential for transmitting signals securely through synchronized chaotic dynamics [1]. Building on these principles, numerous encryption techniques have been proposed, employing chaotic systems to encode and decode messages, with certain success in the encryption of images [2,3], audio [4], and other data types.

Among the more sophisticated approaches, chaotic oscillators have been utilized to implement secure message-masking methods by generating pseudorandom keys [5], while chaos synchronization in coupled map lattices has been shown to achieve a balance between encryption speed and security [6]. Time-delayed coupling mechanisms have also enabled secure key exchange protocols, reducing the probability of successful synchronization by an attacker [7]. Advances in hyperchaotic systems, such as those described in Ref. [8], have improved encryption, achieving multi-user secure data transmission.

Vulnerabilities in chaos-based cryptographic schemes have also been identified. Flaws such as limited key space, insensitivity to key mismatches, and susceptibility to known-plaintext attacks have been highlighted in Ref. [9], while synchronization-based methods face risks from intruders reconstructing sender dynamics via return maps [10]. Moreover, one-dimensional maps commonly used in cryptographic algorithms suffer from dynamical degradation, mak-

ing them vulnerable to signal estimation techniques and other forms of attack [11-13]. Specific weaknesses in image encryption schemes have been exploited through methods such as chosen-plaintext attacks and genetic algorithm-based decryption [14-16].

This paper revisits the use of Lorenz system synchronization for message transmission. Starting with a monochromatic signal, we analyze how the fidelity of the retrieved signal depends on the frequency of the transmitted message and demonstrate that the method is susceptible to hacking via Fourier spectrum analysis. Furthermore, we extend the study to text message transmission, evaluating the recovery and security of such messages. As an alternative, we propose a method called Plain Convolution Encryption (PCE) for message encryption. We demonstrate that this method not only enhances fidelity but also provides increased resistance to decryption attempts compared to synchronization-based schemes.

The paper is organized as follows. In Sec. 2 we briefly describe the method to send a message and set notation, while in Sec. 3 we evaluate the effectiveness of using synchronization. In Sec. 4 we present the results of the PCE approach and finally in Sec. 5 we draw some conclusions.

2. General transmission

Message encryption.

Consider the dynamical system in the form

$$\begin{cases} \dot{\vec{x}} = \vec{f}(t, \vec{x}; \vec{\alpha}), \\ \vec{x}(0) = \vec{x}_0, \end{cases} \quad (1)$$

where $\vec{x} = (x_1, x_2, \dots, x_k)$ is a vector of k components x_i with $i = 1, 2, \dots, n$, which are real functions of time t , that represent the solution functions of the system, and $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ are the parameters of the system. In general, our system has the following properties: 1) \vec{f} is a non-linear function, 2) the system is at least three-dimensional, $n \geq 3$, and 3) the solution of the system $\vec{x}(t)$ is sensitive to the initial conditions \vec{x}_0 . With these properties, the dynamical system can exhibit chaotic behavior. Chaotic trajectories $\vec{x}(t)$ resulting from the numerical solution of system (1), can be used to encrypt a message $m(t)$ difficult to decode because it is screened within the chaotic signal. The encryption process involves defining an encryption function g that takes the message and the chaotic trajectory and gives the encrypted message:

$$m_e(t) = g(m(t); \vec{x}), \quad (2)$$

while a requirement for this function is to be invertible in order to allow the recovery of the original message,

$$m(t) = g^{-1}(m_e(t); \vec{x}). \quad (3)$$

Thus, when the receiver reads the encrypted message $m_e(t)$, the only way to decrypt it is by knowing the decryption key, which is $\text{key}_c = \{\vec{x}_0, \vec{\alpha}, \text{model}, \text{encryption function}, \text{method}, \Delta t\}$. Here \vec{x}_0 and $\vec{\alpha}$ are initial conditions and a set of parameters, the “model” is the particular set of equations that compose the system, “method” refers to the numerical method used to integrate the nonlinear system and uses resolution Δt .

Chaos Synchronization for Message Transmission.

Chaos synchronization occurs when two systems, starting from different initial conditions, evolve to follow the same chaotic trajectory. In this work, we use complete synchronization, where the state variables of the receiver system fully match those of the emitter after a transient period. The implementation defines an emitter system, in our case the Lorenz system with certain parameters, and a receiver system that

can synchronize with the emitter. The emitter adds a message embedded within one of the solution functions of the chaotic solution, and the receiver, synchronized with the emitter, recovers the message by subtracting its own chaotic trajectory, as described in Ref. [17].

Specifically, if the chaotic signal from the emitter is one of the components solutions of system (1) called $u(t)$, we use addition as encryption function, so that the emitted signal is $m_e(t) = u(t) + m(t)$. Later on the receiver generates its own $u_r(t)$, and recovers the message with the subtraction

$$m_r(t) = m_e(t) - u_r(t). \quad (4)$$

Due to the properties of chaos synchronization, the key to decrypt the message in this case is $\text{key}_{cs} = \{\vec{\alpha}, \text{model}, \text{method}, \Delta t\}$, where the encryption function is omitted from the key, because in chaos synchronization the message is added to the chaotic signal. We will show that this encryption function is not very safe nor accurate.

3. Use of synchronization with the Lorenz system

3.1. Workhorse example of sending a message

The model we use is the Lorenz system:

$$\begin{aligned} \dot{u} &= a(v - u), \\ \dot{v} &= ru - v - uw, \\ \dot{w} &= uv - bw, \end{aligned} \quad (5)$$

in the chaotic regime, for which we set the parameters to $(a, b, r) = (10, 8/3, 28)$, with initial conditions $(x_0, y_0, z_0) = (5, 5, 5)$. Figure 1 shows the numerical solution of $u(t)$ in the interval $t \in [0, 200]$ with a resolution $\Delta t = 0.001$. The numerical solution is calculated using a standard RK4 method.

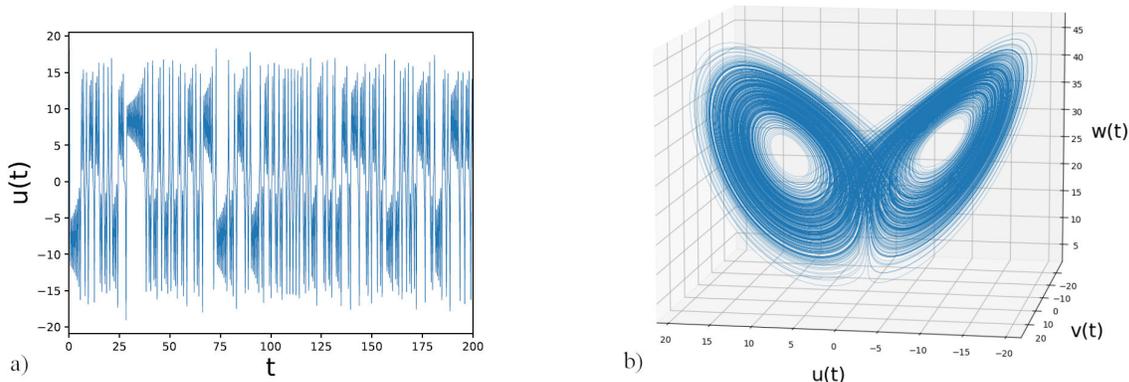


FIGURE 1. Solution of the Lorenz system in the chaotic regime for the initial conditions $(u_0, v_0, w_0) = (5, 5, 5)$ and parameters $(a, b, r) = (10, 8/3, 28)$. This solution is constructed using resolution $\Delta t = 0.001$ over the interval $t \in [0, 200]$ and the RK4 integrator. The time-series of $u(t)$ and the phase space trajectory of the system are shown.

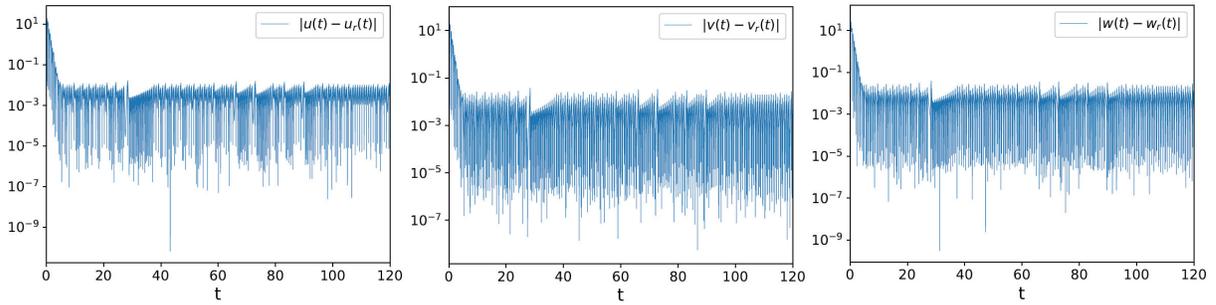


FIGURE 2. Time series showing the synchronization of the emitter and the receiver solutions ($u(t), v(t), w(t)$) and ($u_r(t), v_r(t), w_r(t)$) in a time window of $t \in [0, 120]$. Notice that the message $m(t)$ has a Gaussian envelope centered at $t = 100$, a way after the synchronization of the system happens.

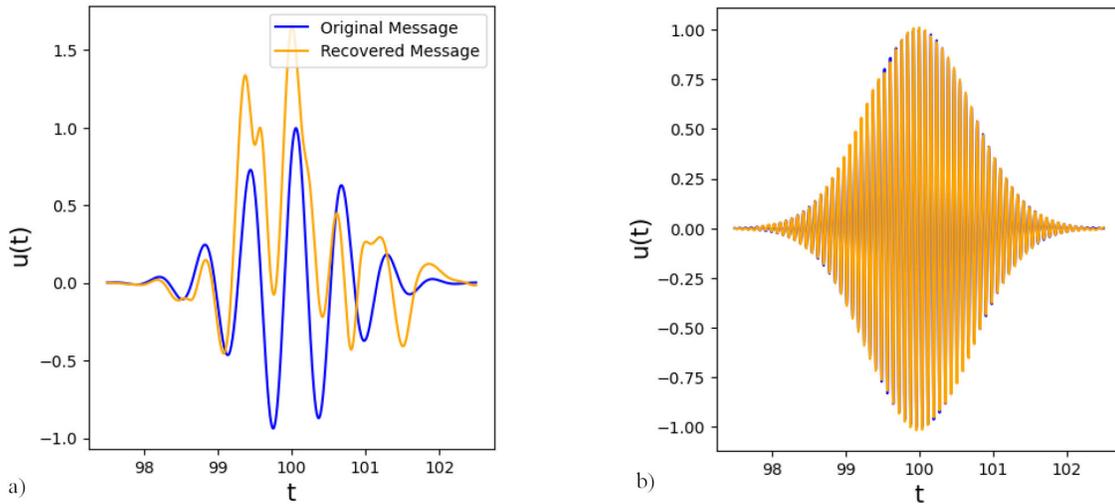


FIGURE 3. Comparison between $m(t)$ and $m_r(t)$ for $A = 1$ and two different frequencies, a) $\omega = 10$ and b) $\omega = 80$.

As an elementary example, we use for emission and recovery a monochromatic function modulated with a Gaussian:

$$m(t) = A \sin(\omega t) e^{-(t-100)^2}, \quad (6)$$

of amplitude A and frequency ω , centered at $t = 100$.

3.2. Synchronization and message recovery

The receiver consists of a system defined with the following modified Lorenz system:

$$\begin{aligned} \dot{u}_r &= a(v_r - u_r), \\ \dot{v}_r &= r m_e(t) - v_r - m_e(t) w_r, \\ \dot{w}_r &= m_e(t) v_r - b w_r, \end{aligned} \quad (7)$$

where the principal characteristic is that $m_e(t)$ in the right hand sides of the equations, is the solution of the emitter system plus the sent message. Synchronization allows to use a key such that $\vec{\alpha}$ contains the parameters a, b, r and there is no need to include the initial conditions in the key, an advantage of the synchronization based method. Thus, for the receiver system we use the initial conditions $(u_{r0}, v_{r0}, w_{r0}) =$

$(25, 6, 50)$, which are different from those used by the emitter since they are not in the key, while the parameters of the emitter $(a, b, r) = (10, 8/3, 28)$ and the resolution $\Delta t = 0.001$ are part of the key, that we use to calculate the solution of (7).

In order to calibrate the numerical solution of emitter and receiver, we test that synchronization happens without the encrypted message, thus the emitted signal is only the solution function u (with $m(t) = 0$) of the Lorenz system (5). As the time evolves, the solution of the receiver u_r, v_r, w_r , start to tend toward the solution of the emitter u, v, w as shown Fig. 2. When this happens, the two systems are said to be synchronized.

Now that we know that synchronization happens, we send the message in a time window where the systems are synchronized using the encryption function $m_e(t) = u(t) + m(t)$. With this signal $m_e(t)$, the solutions u_r, v_r, w_r of the receiver are calculated, which, as seen before, tend to u, v, w . Then the message $m_r(t)$ can be recovered using the subtraction in Eq. (4).

Fidelity and error.

The message reconstructed by the receiver $m_r(t)$ is not exactly the original one $m(t)$. The differences between these

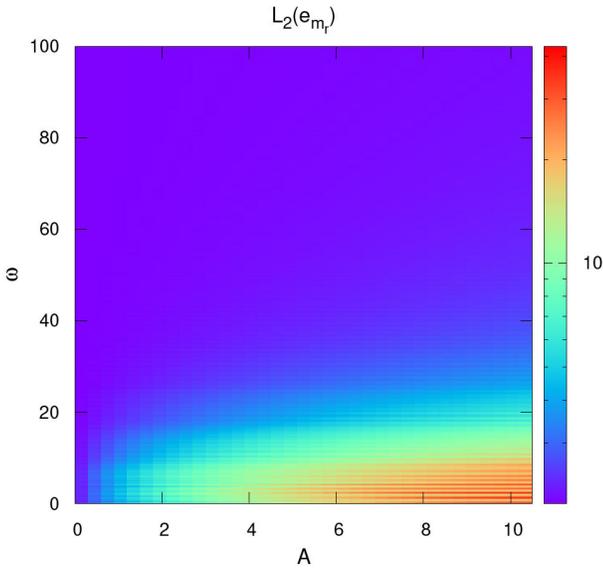


FIGURE 4. The L_2 norm of $e = m(t) - m_r(t)$ for a set of messages $m(t)$ of type (6) with parameters in the range $A \in [0.1, 11]$ and $\omega \in [0.1, 100]$. This plot illustrates how the fidelity in the message recovery degrades for small frequencies and big amplitudes.

two are shown in Fig. 3 for two values of ω . At first sight one can guess that the differences are bigger/smaller for small/high frequencies. In order to illustrate how fidelity depends on A and ω , we calculate the L_2 norm of the error

in the recovered signal $e = m(t) - m_r(t)$, which is summarized in Fig. 4. The errors are bigger for small frequencies and big amplitudes of this monochromatic message. This is a considerably simple message, but helps illustrating how fidelity degrades in terms of the properties of the signal itself. However, this simple case reflects the quality of $m_r(t)$.

Figure 5 shows the signal of $u(t)$ and $m_e(t) = u(t) + m(t)$ in a neighborhood where the Gaussian is maximum for $\omega = 10$ and 80 . Additionally, the corresponding Fourier Transform (FT) calculated in the whole time domain of the solution is included. For the control case with $m(t) = 0$, the solution of the Lorenz system along with its FT is shown, which indicates that the signal has more power in the low frequency domain. The second and third rows of Fig. 5 show the signal and FT of the cases with $\omega = 10$ and $\omega = 80$, respectively. In the later case a high frequency mode can be seen in time and frequency domains, revealing that there is something anomalous in the time-series, possibly a signal.

Here is the dilemma. A message sent with small frequencies hides very well the message within the chaotic signal, as seen in the Fourier Spectra of Fig. 5, unfortunately the fidelity of $m_r(t)$ is not very good as illustrated in Fig. 3. On the other hand, the fidelity of the recovered message is good when using high frequencies as seen in Fig. 3, however, a FT of the signal reveals a clearly identifiable glitch in the tail region of the Fourier Spectrum in Fig. 5, that can be associated to a message.

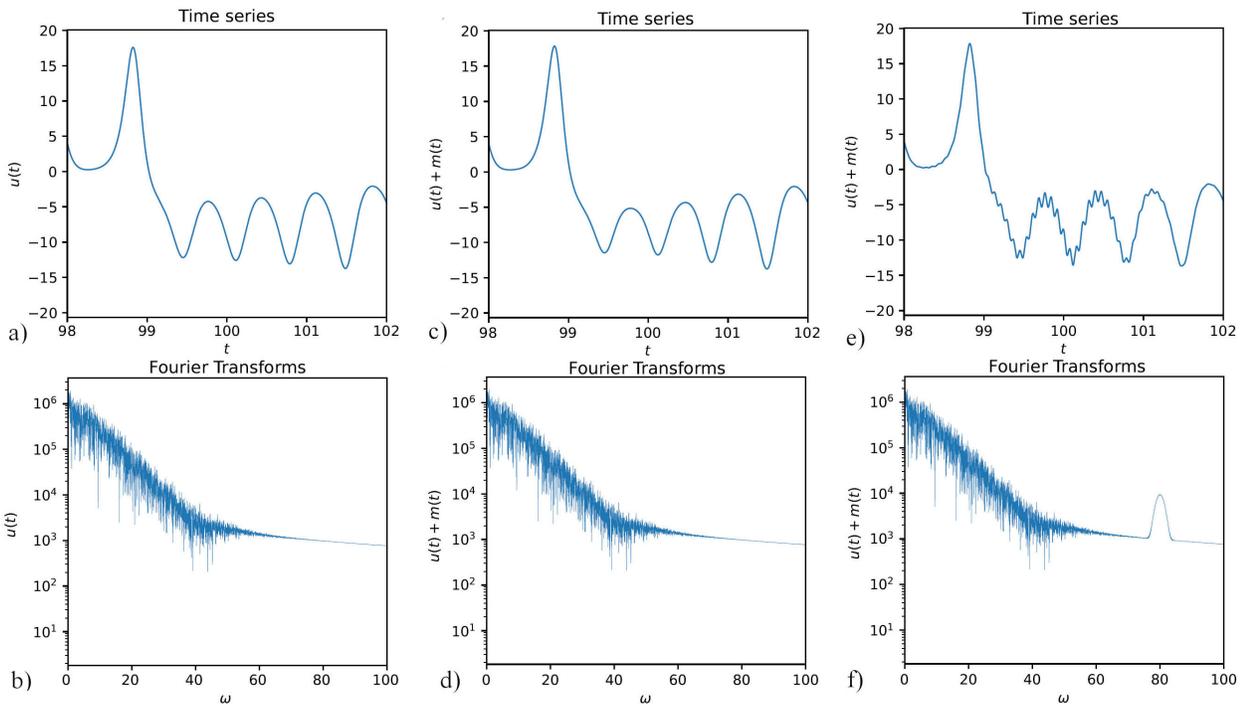


FIGURE 5. Time series of $u(t) + m(t)$ along with its Fourier Transform of three cases, the pure chaotic signal case $m(t) = 0$ (top), the low frequency message (6) and $\omega = 10$ (middle) and the high frequency message with $\omega = 80$ (bottom), both with $A = 1$.

3.3. Hacking this simple signal

If a feature within the high frequency part of the spectrum is detected, the message can be decrypted without needing the key c_s . The first step is to view the transmitted signal in the Fourier space. Since the message can be identified with a glitch in the tail zone, the message modes can be clearly detected even if they cannot be seen in the time series, as shown in Fig. 5 for the case with $\omega = 80$.

The second step is to remove frequencies similar to those of the solution $u(t)$. This is achieved using a high-pass filter with a cutoff frequency ω_{co} , beyond which the message modes should be present. The third step is to apply the inverse Fourier transform to the filtered signal to retrieve the message as follows

$$m_h(t) = \mathcal{F}^{-1} [HPF(\mathcal{F}[u_r(t)])], \quad (8)$$

$$HPF(f(\omega)) = \begin{cases} 0, & \text{if } |\omega| < \omega_{co}, \\ f(\omega), & \text{if } |\omega| \geq \omega_{co}, \end{cases} \quad (9)$$

where $m_h(t)$ is the hacked message, and HPF a high-pass filter.

The accuracy of the hacked message depends on the choice of the cut-off frequency ω_{co} , a bigger value will give high accuracy, but will lose message information in the low frequency part of the spectrum. For this monochromatic example, the L_2 norm of the error between the message sent and the hacked message, $e_h = m_h(t) - m(t)$, was measured using cutoff frequencies using different values of ω_{co} between $[0, 100]$ and the result is in Fig. 6. All sent messages share the same error for low frequencies, but, as ω_{co} approaches the message frequency ω , the error increases until it becomes constant, as the message is eliminated along with the system's solution. Therefore, a good election is the interval

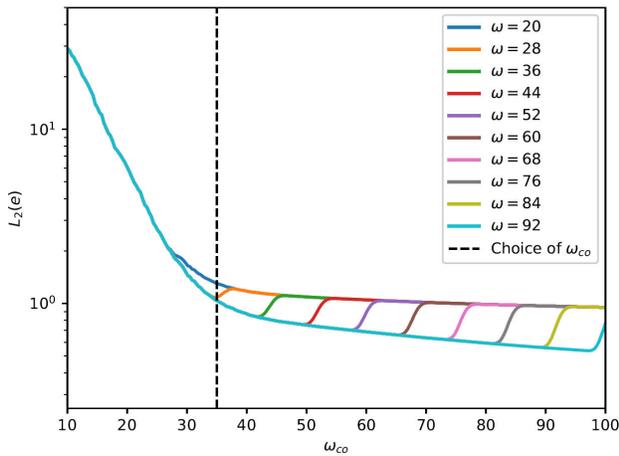


FIGURE 6. L_2 norm of the error $e_h = m_h(t) - m(t)$ for a set of messages with different frequencies and the same amplitude $A = 1$, the vertical black line shows election of ω_{co} . This graph demonstrates the impact of cutoff frequency for $m_h(t)$ as well as the limitation of this method at low frequencies.

$\omega_{co} \in [35, 40]$ because is the region where the interference of the system's solution can be avoided and have a considerable fidelity in the hacked message.

Up to this point we have illustrated how to send, receive and hack a monochromatic message. Thus we show next that more complex messages can also be sent and recovered with synchronization.

3.4. Sending, receiving and hacking text messages

In this section, a text message is translated, loaded, sent, and decrypted using the synchronization of chaos. For this, the following famous text is used:

“Many years later, as he faced the firing squad, Colonel Aureliano Buendía was to remember that distant afternoon when his father took him to discover ice.”

- Gabriel García Márquez, *Cien años de soledad*

Translation of the message.

Each character is converted into an integer using Unicode format according with Table I, arranged as a vector of numbers \vec{m} . In order to hide the message within $u(t)$, we normalize the components of the message as $m_i \rightarrow m_i/300$ to have entries of order one. Then the vector with the message is defined by the following time-series:

$$m(t) = m_i \chi(t_0 + in\Delta t, t_0 + (i+1)n\Delta t), \quad (10)$$

$$\chi(a, b) = \begin{cases} 1, & \text{if } t \in [a, b), \\ 0, & \text{else,} \end{cases} \quad (11)$$

where Δt is the numerical resolution used to calculate $u(t)$, t_0 is the time when the message starts within the time series of $u(t)$, we use n as the number of time steps which separate each character, and finally the label take on values $i = 0, 1, \dots, \text{number of characters} - 1$.

Loading the message.

Once $m(t)$ is generated, the signal is injected within the sender solution $u(t)$ using an Encryption Function. As seen above, there are two parameters used to inject the message, t_0 , that for our particular text message is set to $t_0 = 100$, far beyond synchronization as seen in Fig. 2, and n , that is related to the frequency in which the pieces of the message are sent, a small/big n will correspond to high/low frequencies, that could be captured an FT like in Fig. 5 for the monochromatic message.

Retrieving the message.

Likewise in the monochromatic signal, the message is obtained by subtracting the synchronized solution calculated by the receiver and obtaining m_r using Eq. (4). Later on we un-normalize defining $m_r \rightarrow 300m_r$ and then round each $m_r(t_i)$ to an integer in order to identify back a number with an associated character of Table I.

In Fig. 7, we shown how the receiver can unpack the message for two values of n , noticing that a bigger value has smaller fidelity and viceversa. The text recovered for $n = 1$ is:

TABLE I. Letters of the alphabet and space with their Decimal-Unicode values. The full catalogue including numbers and other characters can be found at [18].

Capital		Lowercase	
letter	Unicode	letter	Unicode
A	65	a	97
B	66	b	98
C	67	c	99
D	68	d	100
E	69	e	101
F	70	f	102
G	71	g	103
H	72	h	104
I	73	i	105
J	74	j	106
K	75	k	107
L	76	l	108
M	77	m	109
N	78	n	110
O	79	o	111
P	80	p	112
Q	81	q	113
R	82	r	114
S	83	s	115
T	84	t	116
U	85	u	117
V	86	v	118
W	87	w	119
X	88	x	120
Y	89	y	121
Z	90	z	122
-	32		

Nboz!zfbst!mbufs- !bt!if!gbdfel!uif!gjsjoh!trvb-
 !Dmpofm!Bvsfmjbop!Cvfodía was to remember
 that distans ‘esdqmnmvgdmghre’ sfcpr-
 mifgkrmbg qamsbof’ b+

and now for $n = 100$:

NainbNPmK · ÍĀĎĪJýúł·šúĹiBĚčýñàĭ Exh<\\
 ĝĪũ žqřũmŷpāl řlŷŷlŷtšLCŮ°zX9:G\qB” Èì°
 iLýoř šÖČöĀ°Ā°ÈĀê Ø

where only a few chains of characters make sense. In both cases, chaos synchronization cannot recover the message with enough fidelity to read the entire text. The percentage of fidelity for $n = 1$ is 20.2% and for $n = 100$ is 0.65% in this particular example.

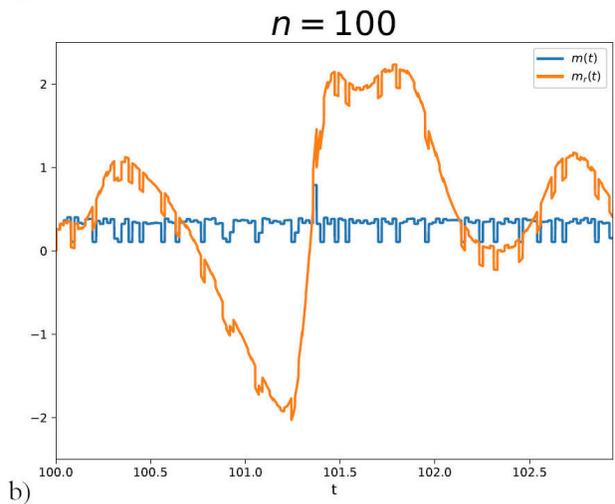
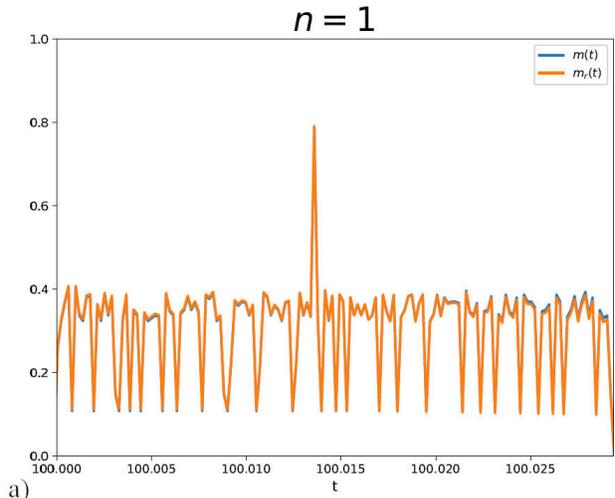


FIGURE 7. Comparison of $m(t)$ and $m_r(t)$ for a) $n = 1$ and b) $n = 100$. The plots show how for large n the difference is considerable. The time window in this plot starts at t_0 and ends at the end of the message.

Hacking the message

For this purpose we use the same method as for the monochromatic signal (8), with a cut-off frequency $\omega_{co} = 35$. The hacked message when using $n = 1$ reads

Nboz!zfbst!mbufs-!bt!ie faced the firing
 squad, Colonel Auqdkh‘mnAtdmci‘v‘rsnqldd
 ladqsg‘schrs‘msafternoon when his father took
 him to!ejtdpwfs!jdf

whereas for $n = 100$ the text is

&v[Saba[tk] 2u2wp(jccee”wmj%jjqehl mtci4)
 Mxtuqfk=qm” gdii :l[cYáUkVimjqfohpdfqoXj[
 bosdt—*)hjs_ g_ [mdhw.yz)g[fUO]jnwz3—{z’s
 hUYdVepcu’uvP

Notice that for $n = 1$ the message is partially hacked with a percentage fidelity of 43.23%, because it is transmitted in a high frequency band, whereas for $n = 100$ the intercepted message has 1.94% . A curious results, not to say ridiculous,

is that the hacked message has smaller errors than the message obtained by the synchronized receiver.

This example shows that also non-trivial messages can be hacked when using chaos synchronization, with limited -but quantifiable- fidelity in terms of the frequency band chosen for transmission. In summary, chaos synchronization used to send a text message is weak because recovery is difficult and it can be partially hacked.

4. Sending a message using Plain Convolution Encryption

In this section, we present the method of PCE for message encryption, with the purpose of addressing the vulnerability at high frequencies in the Fourier spectrum. Unlike in the synchronization based method, the key includes \vec{x}_0 , initial conditions (u_0, v_0, w_0) used to produce the solution of the emitter. This extended key enhances the quality of the recov-

ered message, compared to that in the synchronization based method at low frequencies. In this way, the weight of the security is now on the Encryption Function.

The key is then $\mathbf{key}_e = \{\vec{x}_0, \vec{\alpha}, \text{model, encryption function, method, } \Delta t\}$. We illustrate the method with the Lorenz system (5) like before, with parameters $(a, b, r) = (10, 8/3, 28)$, and initial conditions $(u_0, v_0, w_0) = (5, 5, 5)$. The numerical solution $u(t)$ of the emitter is calculated in the interval $t \in [0, 600]$, using resolution of $\Delta t = 0.0001$ and the RK4 integrator.

For the encryption we use two different functions. The first one is the addition:

$$m_e(t) = m(t) + u(t), \tag{12}$$

and the second function is the convolution:

$$m_e(t) = u(t) * m(t) = \mathcal{F}^{-1} [\mathcal{F}(u(t)) \mathcal{F}(m(t))]. \tag{13}$$

$$m_e(t) = u(t) + m(t)$$

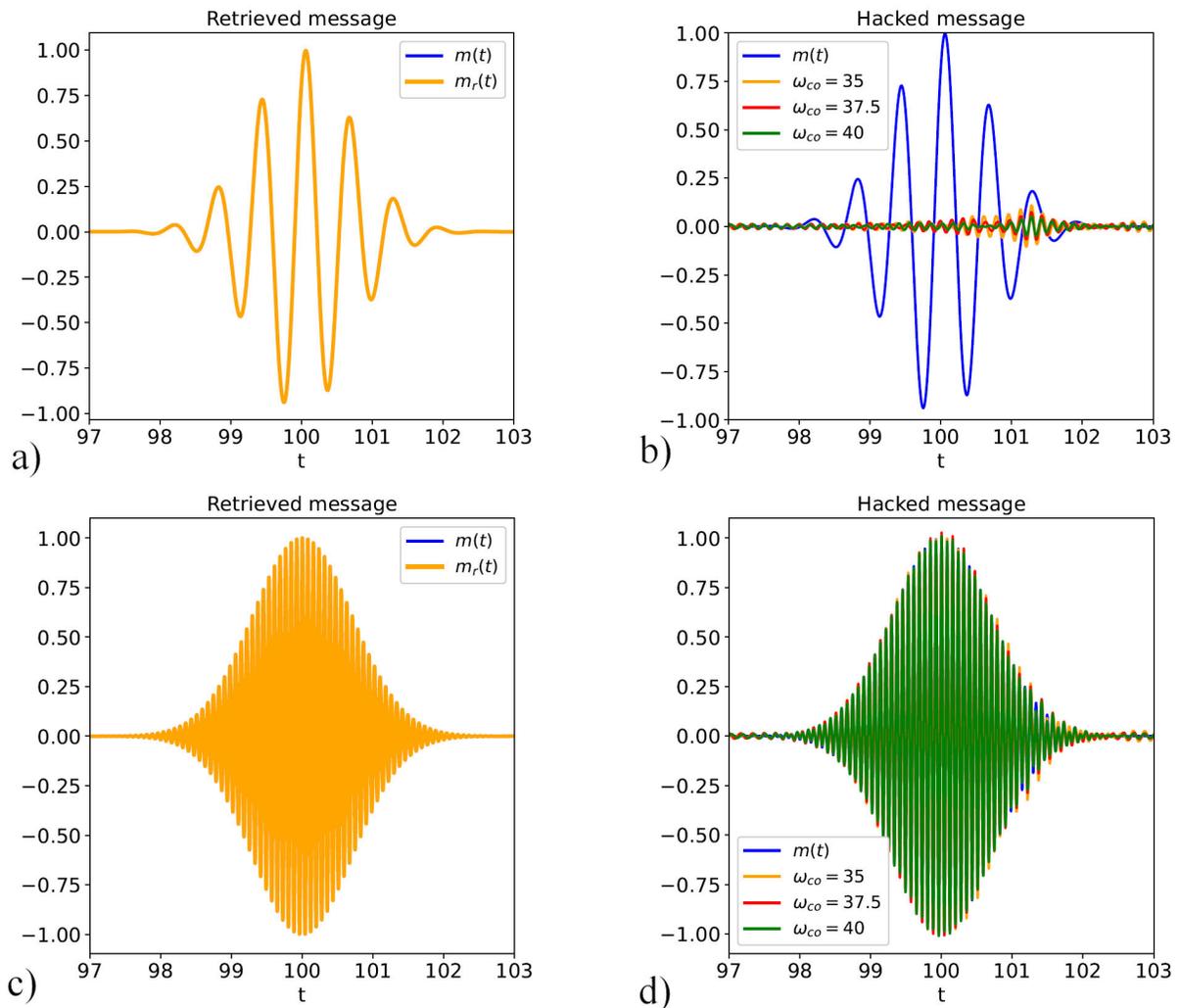


FIGURE 8. Comparison of $m(t)$ and $m_r(t)$ (left), and $m(t)$ and $m_h(t)$ using various cut-off frequencies (right), using the encryption function (12). The sent messages are monochromatic signals (6) with $A = 1$ and two frequencies $\omega = 10$ (top) and $\omega = 80$ (bottom).

$$m_e(t) = u(t) + m(t)$$

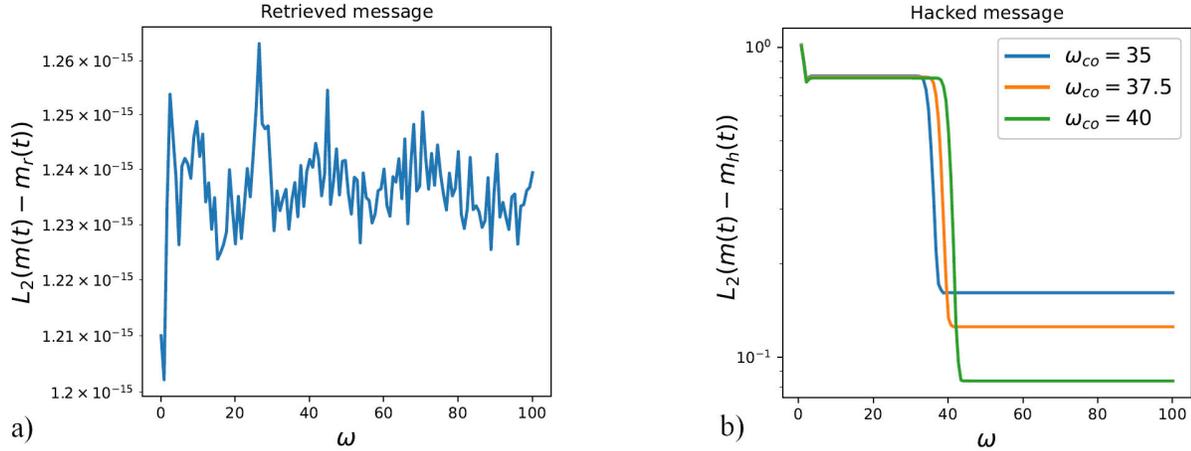


FIGURE 9. a) L_2 norm of $e(t)$ and b) $e_h(t)$. Messages were encrypted using (12). The sent message $m(t)$ is a monochromatic signal defined by (6), with $A = 1$ and $\omega \in [0.1, 100]$. We measure the quality of the hacked message using the method (8) as function of ω with three different values of ω_{co} .

$$m_e(t) = u(t) * m(t)$$

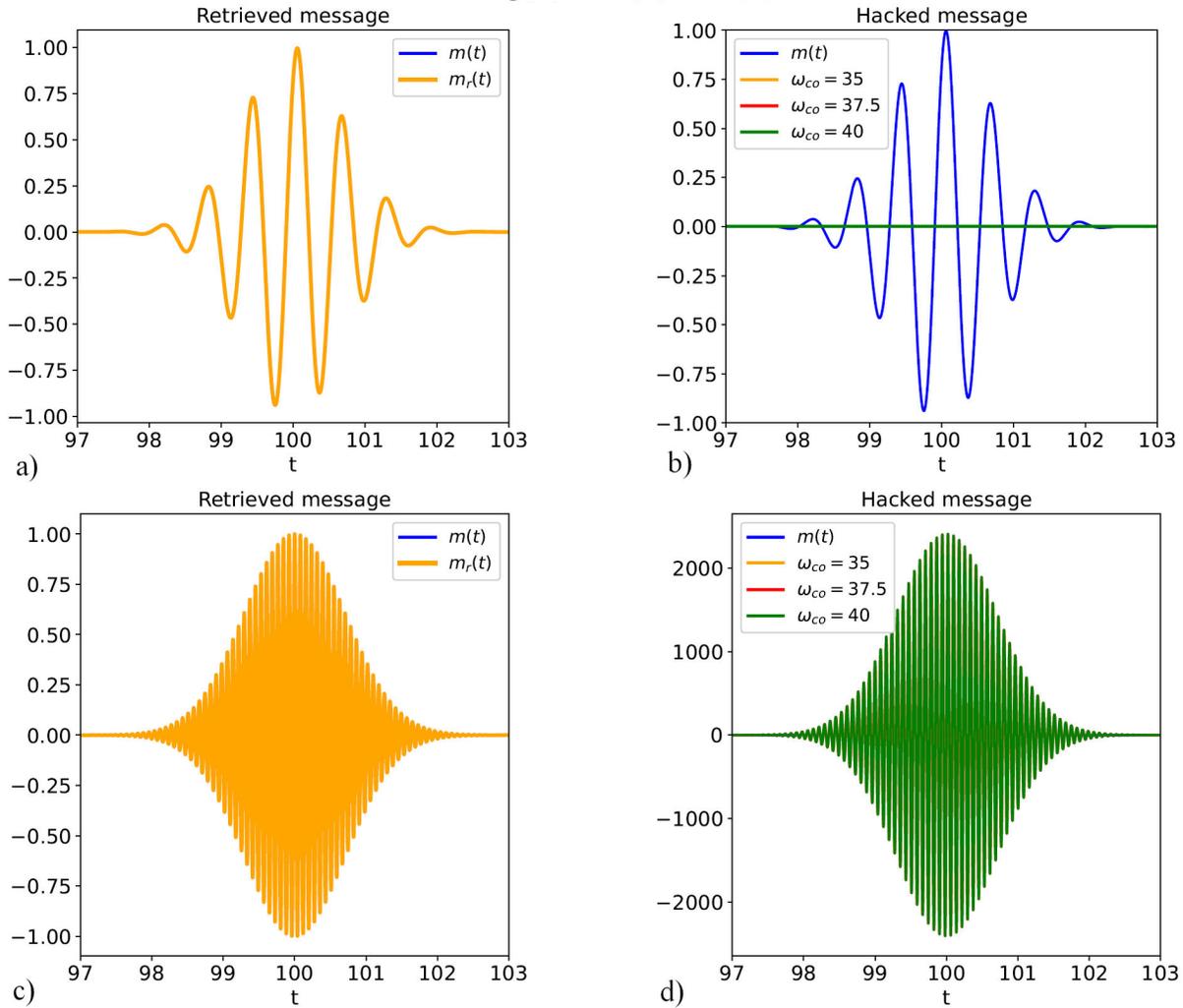


FIGURE 10. At the left the retrieved message $m_r(t)$ and at the right the hacked message $m_h(t)$ using various cut-off frequencies. At the top $\omega = 10$ and at the bottom $\omega = 80$. In this case the message is encrypted with the function (13) and recovered with Eq. (15).

In each case the message is retrieved by the receiver using

$$m_r(t) = m_e(t) - u(t), \quad (14)$$

$$m_r(t) = \mathcal{F}^{-1} \left[\frac{\mathcal{F}(m_e(t))}{\mathcal{F}(u(t))} \right]. \quad (15)$$

For these two functions we test fidelity and security.

4.1. Transmission of a monochromatic wave

We again send the monochromatic signal (6), encrypted in the chaotic signal $u(t)$ generated with the key c .

Retrieving the message. For this we apply the inverse encryption functions (14) or (15), to the encrypted message $m_e(t)$, keeping in mind that the $u_r(t)$ and $u(t)$ are practically the same, because they are generated with the same initial conditions. We present the results for each encryption function separately.

4.1.1. Addition as encryption function

We transmit two monochromatic signals using (12) with amplitude $A = 1$ and frequencies $\omega = 10$ and $\omega = 80$ and the recovered message can be seen in Fig. 8. Since the key contains the initial conditions the retrieved message $m_r(t)$ is practically the same as the sent one $m(t)$. We calculate the L_2 norm of $e(t) = m_r(t) - m(t)$ for $\omega \in [0.1, 100]$, which is rather insensitive to the frequency as seen in Fig. 9.

Hacking the Message.

Figure 8 shows the results of applying the hacking method (8) using cutoff frequencies $\omega_{co} = 35, 37.5$, and 40 . As discussed in the previous section, the low-frequency message proves more challenging to hack with high accuracy, whereas the high-frequency message is intercepted more effectively. The quality of the intercepted message is quantified by the L_2 norm of e_h in Fig. 9, which reinforces the earlier

observation: low-frequency messages are harder to intercept with high fidelity.

4.1.2. Convolution as encryption function

We transmit now the same monochromatic messages, however this time using Eq. (13) as the encryption function. The message $m_r(t)$ is recovered using Eq. (15) and shows high quality as seen in the left column of Fig. 10 for both frequencies $\omega = 10$ and $\omega = 80$. The quality of $m_r(t)$ as function of the frequency is monitored with the L_2 norm of the error in Fig. 11, showing that the quality improves with frequency, although the norm of the error is or round-off error in the whole range.

Hacking the message.

We attempted to hack the message and the result shown in Fig. 11, where the error of $m_h(t)$ is comparable to that using synchronization for low frequencies, however it is two orders of magnitude bigger for high frequencies. For higher frequencies the message is amplified, $m_h(t)$ has the appropriate functional form but not the correct amplitude, as seen in Fig. 10.

These results can be summarized as follows. Using PCE enhances fidelity compared to the use of synchronization. For monochromatic messages, fidelity improves with the extended key: however, since the encryption method remains the same, the security issues persist if we use the encryption function (12). However, the encryption function (13) enhances security by preventing complete recovery of the message from the hacked signal.

4.2. Sending a text message

We now test how PCE works for a text message. We encrypt the same message in the same way as before, and the message is encrypted using the functions (12) and (13).

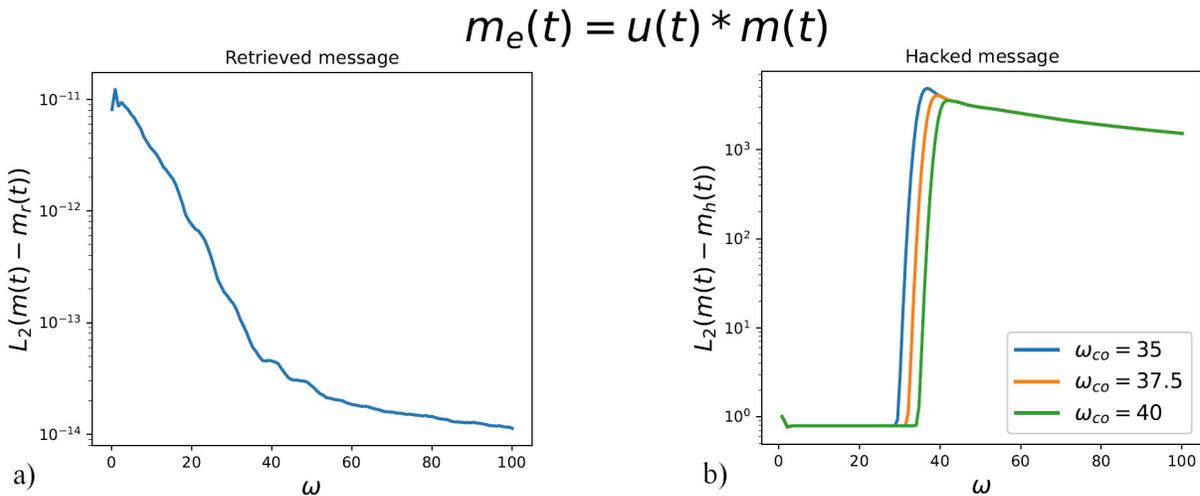


FIGURE 11. We show the L_2 norm of a) e_r and b) e_h as function of ω , when the encryption function is Eq. (13), hacked messages were calculated using three different values of ω_{co} . These results indicate that fidelity is good in the whole range of frequencies, while the quality of the hacked signal worsens for high frequencies. The later is a behavior different from that using synchronization, where the signal was easy to hack in the high frequency region.

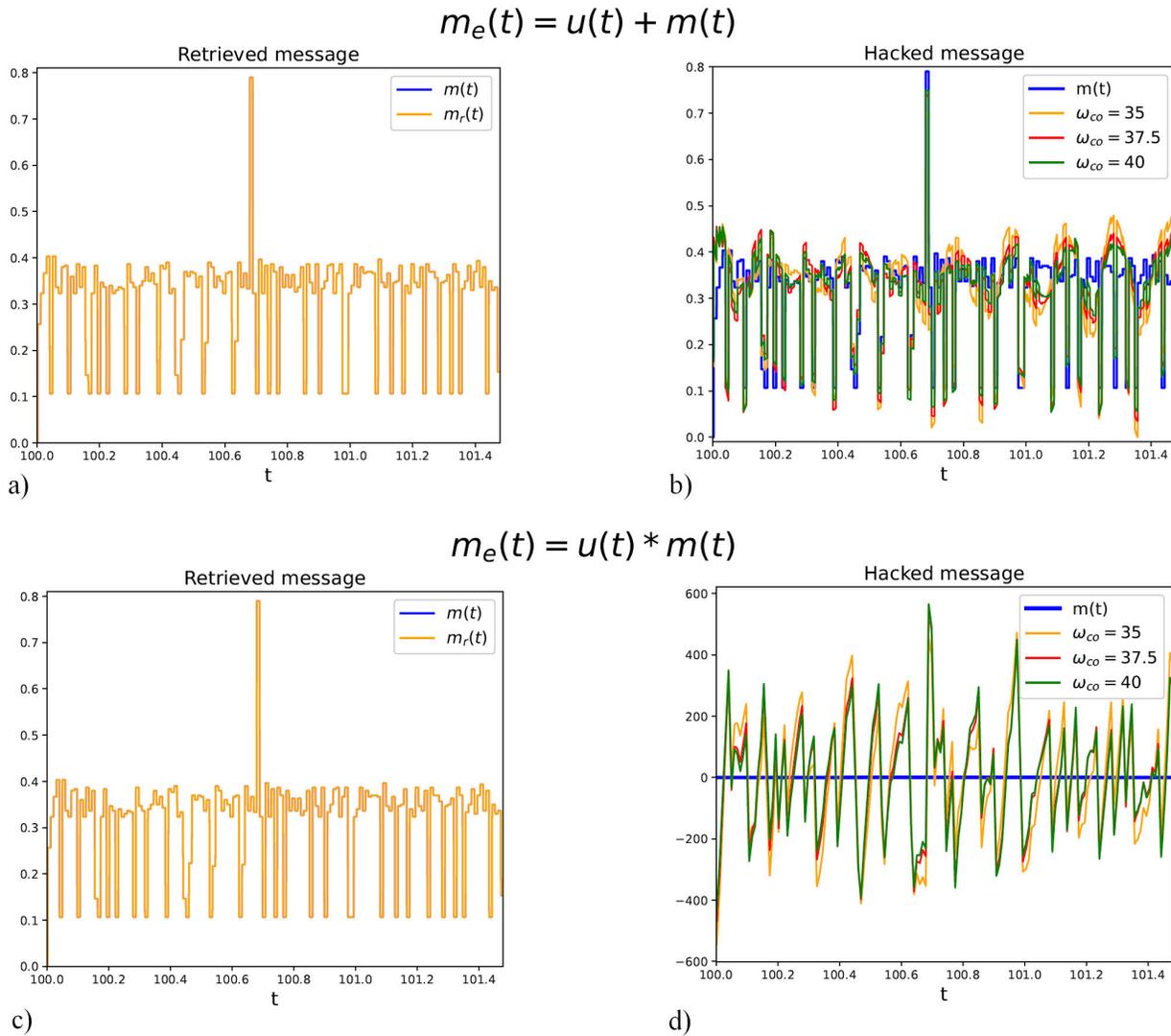


FIGURE 12. Retrieved message $m_r(t)$ and hacked message $m_h(t)$ using various cut-off frequencies using the addition encryption Eq. (12) a) and convolution Eq. (13) b). The message sent is the same text used in Subsec. 3.4. with the lapse between characters $n = 100$ and $t_0 = 100$.

The message is retrieved by the receiver with the key c and the inverse encryption functions (14) and (15). The retrieved messages $m_r(t)$ are shown at the left of Fig. 12 for the two Encryption Functions. The PCE has 100% fidelity for the text message, in the sense that there are no errors in any character of the text.

Hacking the message.

The upper-right graph of Fig. 12 presents the results of the hacked message $m_h(t)$ obtained using method (8) with different values of the cutoff frequency. The results show at least 95% error across the tested frequencies, as only a few characters of the original message were correctly recovered.

For the Encryption Function with convolution (13) we do not recover any character correctly when attempting to hack the message, even some parts of the hacked message $m_h(t)$ are negative as seen in bottom right graph Fig. 12. These results correspond to $n = 100$, which represents a low fre-

quency message and is therefore more difficult to hack by inspecting the FT. As previously discussed, for $n = 1$, the results in terms of quality of $m_r(t)$ and the hackability of $m_h(t)$ are similar. Given the security is poor for $n = 1$, it is not the ideal choice for the value of n .

5. Conclusions

In this work, message transmission and encryption using chaotic systems were revisited, comparing the performance of synchronization-based methods with PCE. The study focuses on the reliability of message recovery and the susceptibility of these methods to attacks using Fourier analysis.

For synchronization-based encryption using the Lorenz system, a trade-off was observed between message fidelity and vulnerability. Low-frequency monochromatic messages were effectively immersed within the chaotic signal, as re-

flected in the Fourier spectrum, but exhibited poor recovery fidelity. In contrast, high-frequency messages were more easily recovered, but introduced identifiable artifacts in the frequency domain, making them susceptible to hacking. When transmitting text messages, synchronization techniques displayed significant degradation in recovery fidelity, particularly as the time step between characters increases, leading to incomplete and errors in the message retrieval.

The Fourier-based hacking method proved highly effective in identifying and partially reconstructing messages. In cases where synchronization failed to recover the message accurately, hacking techniques often outperformed the synchronized receiver, exposing a fundamental weakness of the synchronization based method.

To address these vulnerabilities, Plain Convolution Encryption was introduced using both addition and convolution as encryption functions. The addition-based approach improved message recovery fidelity but remained susceptible to hacking, especially for high-frequency signals. In contrast, the convolution-based method substantially enhanced both message fidelity and resilience to hacking attempts. While monochromatic messages encrypted with convolution are successfully recovered, the hacking process cannot recover the original amplitude. For text messages, the convolution-based method achieved 100% recovery fidelity and proved resistant to hacking using the Fourier analysis, even in low-frequency transmissions.

In conclusion, the results emphasize the limitations of

synchronization-based encryption for secure communication and highlight the advantages of PCE, particularly when employing convolution-based encryption. This method provides a considerable improvement in both message fidelity and security. Future advancements in encryption strategies will be required to address the vulnerabilities exposed by spectral analysis techniques and further enhance the robustness of chaos-based encryption systems.

Appendix

A. Converting a List of Characters to a Vector of Integers

Suppose we have a string of characters that we want to encrypt. Each character can be identified with an integer according to the Unicode (Decimal) format, as shown in Table I for an example alphabet. We can convert the string into a list of integers, which acts like the discrete message $m_i = m(t_i)$ for some values of t_i . For example, if the string is "Hello World", the discrete message is $m = \{72, 101, 108, 108, 111, 32, 87, 111, 114, 108, 100\}$.

Acknowledgments

This research is supported by grants CIC-UMSNH-4.9 and Laboratorio Nacional de Cómputo de Alto Desempeño Grant No. 1-2024.

-
1. U. Parlitz, L.O. Chua, Lj. Kocarev, K.S. Halle, and A. Shang, Transmission of digital signals by chaotic synchronization, *International Journal of Bifurcation and Chaos* **02** (1992) 973, <https://doi.org/10.1142/S0218127492000562>.
 2. O. M. Al-Hazaimeh, M. F. Al-Jamal, N. Alhindawi, and A. Omari, Image encryption algorithm based on lorenz chaotic map with dynamic secret keys, *Neural Computing and Applications* **31** (2019) 1, <https://doi.org/10.1007/s00521-017-3195-1>.
 3. D. A. Q. Shakir, A. Salim, S. Q. A. Al-Rahman, and A. M. Sagheer, Image encryption using lorenz chaotic system, *Journal of Techniques* **5** (2023) 122, <https://doi.org/10.51173/jt.v5i1.840>.
 4. T. Haridas, Upasana S.D., Vyshnavi G., M. S. Krishnan, and S. Shankar Muni, Chaos-based audio encryption: Efficacy of 2d and 3d hyperchaotic systems, *Franklin Open* **8** (2024) 100158, <https://doi.org/10.1016/j.fraope.2024.100158>.
 5. B. Fraser, P. Yu, and T. Lookman, Steps towards improving the security of chaotic encryption, *Phys. Rev. E* **66** (2002) 017202, <https://doi.org/10.1103/PhysRevE.66.017202>.
 6. S. Wang, J. Kuang, J. Li, Y. Luo, H. Lu, and G. Hu, Chaos-based secure communications in a large community, *Phys. Rev. E* **66** (2002) 065202, <https://doi.org/10.1103/PhysRevE.66.065202>.
 7. E. Klein, R. Mislovaty, I. Kanter, and W. Kinzel, Public-channel cryptography using chaos synchronization, *Phys. Rev. E* **72** (2005) 016214, <https://doi.org/10.1103/PhysRevE.72.016214>.
 8. D. Rontani, M. Sciamanna, A. Locquet, and D. S. Citrin, Multiplexed encryption using chaotic systems with multiple stochastic-delayed feedbacks, *Phys. Rev. E* **80** (2009) 066209, <https://doi.org/10.1103/PhysRevE.80.066209>.
 9. S. Li, G. Alvarez, G. Chen, and X. Mou, Breaking a chaos-noise-based secure communication scheme, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **15** (2005), <https://doi.org/10.1063/1.1856711>.
 10. S. Bu and B.-H. Wang, Improving the security of chaotic encryption by using a simple modulating method, *Chaos, Solitons and Fractals* **19** (2004) 919, [https://doi.org/10.1016/S0960-0779\(03\)00260-1](https://doi.org/10.1016/S0960-0779(03)00260-1).
 11. G. Alvarez and S. Li, Cryptanalyzing a nonlinear chaotic algorithm (nca) for image encryption, *Communications in Nonlinear Science and Numerical Simulation* **14** (2009) 3743, <https://doi.org/10.1016/j.cnsns.2009.02.033>.

12. J. S. Teh, M. Alawida, and Y. C. Sii, Implementation and practical problems of chaos-based cryptography revisited, *Journal of Information Security and Applications* **50** (2020) 102421, <https://doi.org/10.1016/j.jisa.2019.102421>.
13. M. Alawida, Enhancing logistic chaotic map for improved cryptographic security in random number generation, *Journal of Information Security and Applications* **80** (2024) 103685, <https://doi.org/10.1016/j.jisa.2023.103685>.
14. B. Norouzi and S. Mirzakuchaki, Breaking an image encryption algorithm based on the new substitution stage with chaotic functions, *Optik* **127** (2016) 5695, <https://doi.org/10.1016/j.ijleo.2016.03.076>.
15. S. Noshadian, A. Ebrahimzade, and S.Javad Kazemitabar, Breaking a chaotic image encryption algorithm, *Multimedia Tools and Applications* **79** (2020) 25635, <https://doi.org/10.1007/s11042-020-09233-6>.
16. R. Zhou and S. Yu, Break an enhanced plaintextrelated chaotic image encryption algorithm, *Chaos, Solitons and Fractals* **181** (2024) 114623, <https://doi.org/10.1016/j.chaos.2024.114623>.
17. K. M. Cuomo and A. V. Oppenheim, Circuit implementation of synchronized chaos with applications to communications, *Phys. Rev. Lett.* **71** (1993) 65, <https://doi.org/10.1103/PhysRevLett.71.65>.
18. The Unicode Consortium, Unicode character code charts, (2024).