BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **22**, No 4
Sofia • 2022 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2022-0037

An Approach for Securing JSON Objects through Chaotic Synchronization

Omar S. Gómez[1], *Raúl H. Rosero*[1], *Juan C. Estrada-Gutiérrez*[2], *Maricela Jiménez-Rodríguez*[2]

[1]GrIISoft Research Group, Escuela Superior Politécnica de Chimborazo (ESPOCH), Riobamba, Chimborazo 060155 Ecuador
[2]Centro Universitario de la Ciénega, Universidad de Guadalajara, Ocotlán, Jalisco 48350 México
E-mail: ogomez@espoch.edu.ec

***Abstract*: *Nowadays the interoperability of web applications is carried out by the use of data exchange formats such as XML and JavaScript Object Notation* (*JSON*)*. Due to its simplicity, JSON objects are the most common way for sending information over the HTTP protocol. With the aim of adding a security mechanism to JSON objects, in this work we propose an encryption approach for cipher JSON objects through the use of chaotic synchronization. Synchronization ability between two chaotic systems offers the possibility of securing information between two points. Our approach includes mechanisms for diffusing and confusing JSON objects* (*plaintext*)*, which yields a proper ciphertext. Our approach can be applied as an alternative to the existing securing JSON approaches such as JSON Web Encryption* (*JWE*)*.*

***Keywords*: *JSON object, chaotic synchronization, chaos-based system, encryption, cryptography, Rössler non-linear system.*

1. Introduction

Among the software web development community, JavaScript Object Notation (JSON) objects are commonly used for sending and retrieving information. JSON [1, 2] is used as a simple lightweight object serialization approach or data format which is based on the JavaScript programming language syntax. At a glance, JSON objects can be seen as dictionaries consisting of key-value pairs. An example of a JSON object structure is the following:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 27,
  "city": "New York"
}
```

Because of JSON simplicity, i.e., it is easily readable either by humans or by machines, JSON is one of the most popular formats for exchanging data on the

Web [3]. Nowadays most web applications rely on a web service software architecture style where software components (or services) communicate with each other through the use of APIs (Application Programming Interfaces). In this regard, JSON objects are the predominant interchange format for sending API requests and responses over the HTTP protocol.

Due to JSON objects being commonly transported over the HTTP protocol, an approach to secure them is through the HTTPS protocol. In this sense, HTTPS offers a point to point transport layer encryption by using the Transport Layer Security (TLS) protocol [4]. However, it is worth noting that TLS becomes insufficient in scenarios where JSON objects are redirected over untrusted proxies (intermediates). In this context, JSON objects can be encrypted to enhance security.

A common approach to secure JSON objects is through the use of JSON Web Encryption [5]. JSON Web Encryption (JWE) is a standard that provides a syntax for the exchange of encrypted data. The implementations of JWE are said to provide authenticated encryption mechanisms aimed at ensuring the confidentiality, authenticity, and integrity of JSON objects.

However, some vulnerabilities have been identified in the implementations of the JWE. One of them could allow attackers to retrieve private keys. It has been found that implementations of JWE are subject to a classic invalid curve attack [6, 7]. Another vulnerability identified is related to the Bleichenbacher Million Message Attack [8].

With the aim of having an alternative to the JWE, in the present work we present an approach for securing JSON objects by means of chaotic synchronization. Chaotic synchronization occurs when two chaotic systems are synchronized with each other under the driving of a signal which is sent from one system (master scheme) to the other (slave scheme). Under this circumstance, information can be securely transmitted between the master and the slave.

The rest of the document is organized as follows: Section 2 presents an overview of chaotic synchronization and related work. Section 3 presents the approach for securing JSON objects by means of chaos-based synchronization. Section 4 shows a case implementation example of the proposed approach. Section 5 presents a statistical assessment of the proposed approach. Finally, in Section 6 we present the discussion and conclusions.

2. Related work

Chaos or chaotic synchronization is a phenomenon that occurs when two coupled chaotic dynamical systems, after a while, yield the same solution regardless of their initial set of conditions. This phenomenon was firstly described in [9]. H u y g e n s [9] observed a “sympathetic” motion in two pendulum clocks coupled through a wooden structure. Some time after, it was observed that the coupled clocks started to oscillate in consonance and the same direction.

More recently, since the work reported in [10], chaotic synchronization has gained attention from the scientific community. In [10], authors reported that chaotic systems can be synchronized by sending or driving only a portion of the state space information from one system known as the master to another system known as a slave. This finding opened the door to the study and application of securing communication systems, by using the rest of the state space information that is not sent [11-13].

A simple setup for complete synchronization relies on the use of a master system acting as the transmitter, and a slave system acting as the receiver. Two commonly dynamical systems used for chaotic synchronization are the L o r e n z [14] and the R ö s s l e r [15, 16] systems.

The master scheme (either Lorenz or Rössler) has components (vector variables) in three directions; however, one of these directions is used for coupling the master with the slave, this variable is also known as the driver. Once conducted the synchronization between the master and the slave, variables of both systems start to yield the same values, making possible a secure transmission of information.

In the case of the Lorenz system [14], the master scheme is represented by the following three non-linear ordinary differential equations:

(1) $$\dot{x_1} = \sigma(y_1 - x_1),$$

(2) $$\dot{y_1} = x_1(\rho - z_1) - y_1,$$

(3) $$\dot{z_1} = x_1 y_1 - \beta z_1,$$

where x_1, y_1, and z_1 are the initial conditions of the system, and σ, ρ and β are system parameters. Considering x_1 as the driver for coupling the master with the slave, the slave scheme is represented by the following non-linear ordinary differential equations:

(4) $$\dot{y_2} = x_1(\rho - z_2) - y_2,$$

(5) $$\dot{z_2} = x_1 y_2 - \beta z_2.$$

Note that variable x_1 is the same in both schemes, the master and the slave; i.e., variable x is replaced in the slave. On the other hand, using the Rössler system [17], the master scheme is represented by the following three non-linear ordinary differential equations:

(6) $$\dot{x_1} = -y_1 - z_1,$$

(7) $$\dot{y_1} = x_1 + a y_1,$$

(8) $$\dot{z_1} = b + z_1(x_1 - c),$$

where x_1, y_1, and z_1 are the initial conditions whereas a, b and c are system parameters. For example, considering y_1 as the driver for coupling the master with the slave, the slave scheme can be represented by the following equations:

(9) $$\dot{x_2} = -y_1 - z_2,$$

(10) $$\dot{z_2} = b + z_2(x_2 - c).$$

Also in this case, variable y_1 is fully replaced in the slave. In these two systems, synchronization is possible regardless of initial conditions set in variables x, y, z of the system. However, system parameters should be maintained in both schemes (master and slave) to achieve complete synchronization. Because y_1 is used for coupling the master with the slave, variables x_1 and z_1 can be used for transmitting the information. A common approach for sending information consists in adding or masking a message on one of the two remaining chaotic signals [18], thus generating a new signal. Considering x_1 as the variable for sending the message, we have the following equation:

$$s(t) = m(t) + x_1(t), \tag{11}$$

where $s(t)$ is the resulting signal of adding $m(t)$, that is the message to be sent to $x_1(t)$; $x_1(t)$ is the chaotic variable of the master system in a time frame t. For unmasking the message, is used:

$$m(t) = s(t) - x_2(t), \tag{12}$$

where $m(t)$ is the regenerated message obtained from subtracting $s(t)$, that is the information-bearing variable, from $x_2(t)$; $x_2(t)$ is the chaotic variable of the slave system in a time frame t.

In addition to masking [17], other approaches used for securing information in chaos-based synchronization are the use of chaotic regime-switching [18], nonlinear mixing of a given information signal with a chaotic one [19], and the use of modulation [17, 20].

Securing information by means of chaotic synchronization has been mainly implemented in electronic circuits and lasers domains [21-25]. More recently, the Internet of Things (IoT) domain also has been reported its application [26]. Chaotic synchronization has been used for encryption of information such as: audio [27-29], image [30-32], video [32, 33] and text [33, 34]. Concerning textual information, we have not found works proposing the encryption of data exchange formats such as XML or JSON.

3. Proposed approach

Once presented the basic structure of JSON objects, and also discussed the related work. In this section, we present the proposed approach for securing JSON objects by means of chaos-based synchronization.

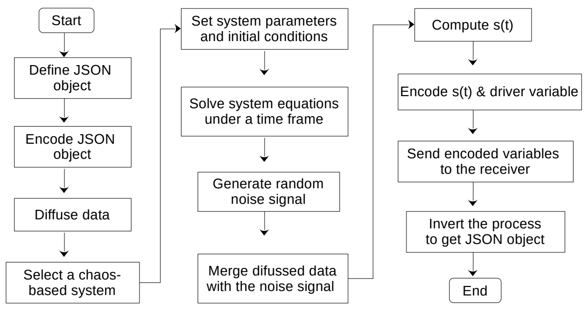
Start
Define JSON object
Encode JSON object
Diffuse data
Select a chaos-based system
Set system parameters and initial conditions
Solve system equations under a time frame
Generate random noise signal
Merge difussed data with the noise signal
Compute s(t)
Encode s(t) & driver variable
Send encoded variables to the receiver
Invert the process to get JSON object
End

Fig. 1. Flowchart of the proposed approach

Given a JSON object, our proposal consists in encoding the JSON object. The encoded JSON object is then diffused by making some operations on its bits. Once diffused the data, a chaos-based system is used for the confusing process. With the selected chaos-based system to work with, the system parameters and the initial conditions are set. The equations of the system are solved under a time frame.

Taking as reference the length of one of the chaotic vector variables a random noise-like binary signal is generated. The diffused data is merged with the binary signal generated, thus having *m*(*t*). The information-bearing signal *s*(*t*) is obtained by masking *m*(*t*) with one of the chaotic vector variables that will be used as the carrier, e.g., *x*(*t*). Once having *s*(*t*) and the chaos vector variable used has the driver, e.g., *y*(*t*), both variables are encoded. The encoded variables are ready to be transmitted to the receiver. The inverse process is conducted at the receiver to retrieve the JSON object. Fig. 1 shows a flowchart of the approach proposed in the present work.

4. Case implementation example

In this section, we describe the implementation of the proposed approach by following the steps presented in Fig. 1 of the previous section. The implementation was done through the Python programming language.

Table 1. Example of diffusion applied over the bits of the encoded JSON object

Encoded JSON object	Encoded JSON object after diffusion
01100101011101110110111101100111	01000011010000110111010110011001
01001001010000110100000101100111	00101101011011010110000101111001
01001001011011010101101001110000	01100001011000110001010100010101
01100011011011100100111000110000	00011001001110011110010110100101
01010100011011010100011001110100	00001101101011010011110101101101
01011010010100110100100100110110	10010011011011010011010101011001
01001001010000110100101001001011	11110011110010011011001101100101
01100010001100100110100001110101	01101001011111010011110101101101
01001001011010010111011101001011	00011001011111010110100100111101
01001001010000110100000101100111	00110001001110011010100101001101
01001001010000110100101001110011	00011001000010011010100101101101
01011001010110000100111000110000	11001001110110010001010101100101
01010100011011010100011001110100	01101001011111010011110101101101
01011010010100110100100100110110	00011001011111010110100100111101
01001001010000110100101001010100	00110001011011010011110101111001
01100010010101110110110000110000	11110011110010010001010110111001
01100001010000110100100101110011	11010101101011010011110101101101
01000011011010010100000101100111	10010011011011010011010110100101
...	...

Assuming we have the following JSON object, we proceed to encode it. The process of encoding allows having an initial confusion of the JSON object:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 27,
  "city": "New York"
}
```

In this example, the encoding is done by using the Base64 format. Base64 is a collection of related encoding designs that represent information in ASCII format. It is commonly used for transferring data over media. It also guarantees that the information stays unchanged during the transferring process. After the encoding step, the following string of bytes is obtained:

ewogICAgImZpcnN0TmFtZSI6ICJKb2huIiwKICAgICJsYXN0TmFtZSI6ICJTbWl0aCIsCiAgICAiYWdlIjogMjcsCiAgICAiY2l0eSI6ICJOZXcgWW9yayIKfQ==

For the next step, we diffuse the bits of the encoded JSON object. The diffusion process is conducted by applying some operations on the bits. Table 1 shows an excerpt of the bits that represent the encoded JSON object, before and after applying diffusion.

The next step consists in selecting a chaos-based system. The selected system is used for the master and the slave schemes. For this example, in the case of the master, we selected the Rössler system with the parameters a=0.2 and c=5.7, in the case of parameter b, we used a value set in one of the chaotic regions of this system. Vector variables x, y, z were initialized with values 0.3, 0.1, and 0.1, respectively. To solve the equations in (6)-(8), we implemented the fourth-order Runge-Kutta method [35, 36] for a fixed time frame. Fig. 2 shows vector variables (x and y) after the system is solved.

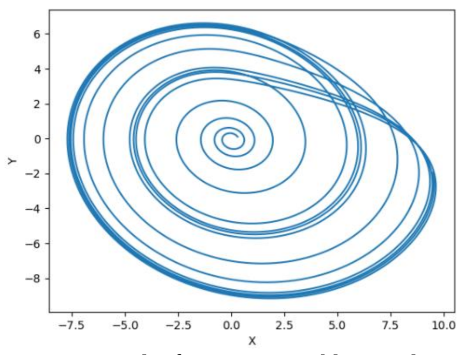

Fig. 2. 2D plot for vector variables x and y

Once solved the equations in the master scheme, and taking as reference the length of the chaos variable used for masking (in this case x), a random noise binary signal (vector variable) is generated. The diffused JSON object is then merged with the random noise vector. This vector is masked with the chaos vector variable x, thus obtaining the information-bearing signal. The random noise vector containing the diffused JSON object and the chaos variable x are shown in Fig. 3.

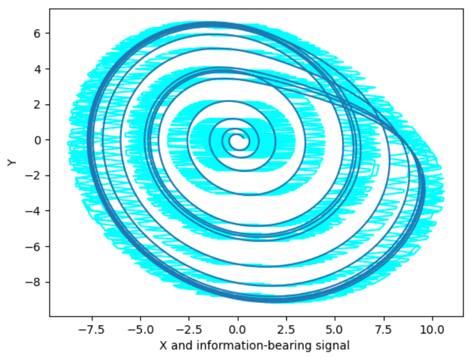

Fig. 3. 2D plot with the chaos variable x (blue) and the information-bearing signal (cyan)

As shown in Fig. 3, the blue color line corresponds to the chaos vector variable *x*. The cyan color line corresponds to the information-bearing signal, *s*(*t*).

The next step consists in encoding the information-bearing variable and the driver chaos variable, in this case, *y*. With te aim of adding an extra layer of confusion, these variables were encoded using the Base64 format, a fragment of the first 50 bytes is next shown:

```
mpmZmZmZuT9jmV6lOTu7P2CBru0l2bw/OvC1jzpzvj8xir9yqQ...
```

The encoded variables are sent to the receiver and the steps previously described are inverted. With the driver and the information-bearing variables decoded, the driver variable is used for coupling the master with the slave system.

In the case of the slave, the same parameters previously set for the master were kept on it. The initial conditions for the slave scheme were set as $x= -2.2$, $y=1.1$, and $z=0.3$. Once resolved the equations in (9) and (10) related to the slave scheme, it is possible to achieve synchronization in both schemes (master and slave). When synchronization occurs, the JSON object can be regenerated by unmasking it from the information-bearing variable. This process is carried out by subtracting the chaos vector variable *x* of the slave scheme from the information-bearing variable. A graphical representation of the synchronization between the information-bearing variable and the vector variable of the slave (x_2) is shown in Fig. 4.

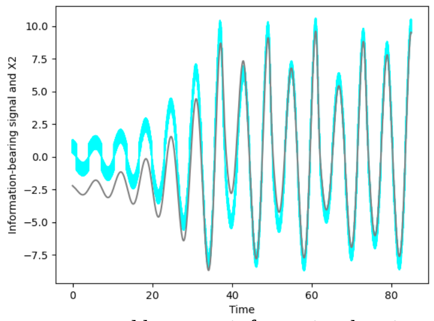

Fig. 4. Synchronization process occurred between information-bearing variable from master (cyan) and chaos variable x_2 from slave (gray)

Once the unmasked process is done, we have the random noise signal that contains the diffused JSON object. The diffused JSON object is extracted, and then the diffusion is removed resulting in the encoded JSON object (in Base64 format). Finally, the JSON object is retrieved after it is decoded.

5. Statistical analysis

In this section, we present a descriptive analysis of the JSON object encrypted with the proposed approach. Fig. 5 shows a correlation plot between the JSON object used as plaintext and the regenerated JSON object after the decryption process.

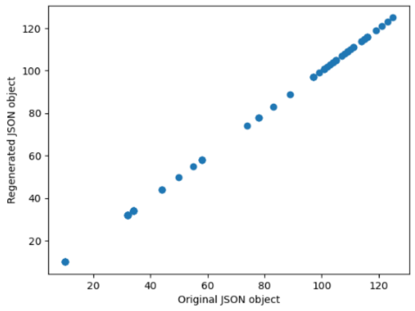

Fig. 5. Correlation plot between original JSON object and the JSON regenerated

As shown in Fig. 5, the original JSON object is completely regenerated (without having loss of information) after the decryption process. A Pearson correlation coefficient of 1.0 was observed. With the aim of assessing a possible relation between the plaintext and the ciphertext, Fig. 6 shows a scatter plot between these two elements.

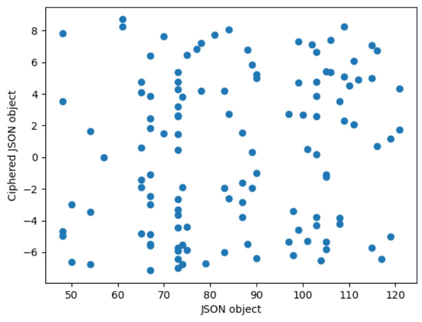

Fig. 6. Correlation plot between plaintext and ciphertext information

As shown in Fig. 6., there is not a clear pattern between the plaintext and the ciphertext. The computed Pearson correlation coefficient was 0.14 with a nonsignificant p-value of 0.11, thus suggesting a lack of relationship between the plaintext and the ciphertext.

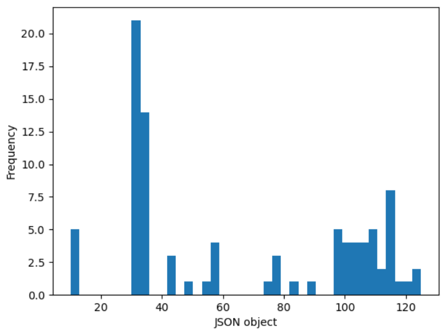

Fig. 7. Histogram of the original JSON object

Another analysis conducted is concerned with assessing the distribution of the original JSON object (plaintext) with regards to the encrypted JSON object (ciphertext). Fig. 7 shows a histogram with the distribution of values representing the plaintext.

As shown in Fig. 7, the higher frequency of values representing the original JSON object is between 20 and 40. It is also observed in regions without any frequency. Regarding the ciphered JSON object, its distribution is shown in Fig. 8.

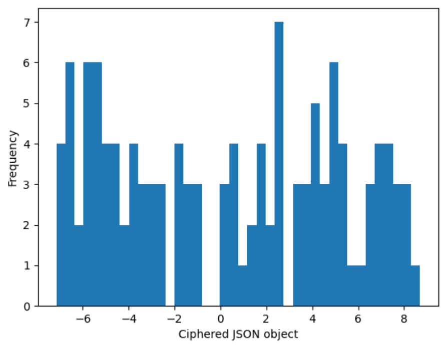

Fig. 8. Histogram of the ciphered information

In the case of Fig. 8, a more uniform distribution is observed. Values of the ciphertext have a major frequency with regards to values of the plaintext, thus suggesting a proper plaintext concealing.

6. Discussion and conclusions

Chaos-based synchronization offers the possibility of sending secure information via chaotic signals. The feasibility of sending JSON objects by means of this approach is described in this work, so this approach can be used as an alternative to existing JSON Web Encryption mechanisms.

The approach presented in Section 3 (see Fig. 1) is enough flexible, its implementation can be tweaked by adjusting its steps of it. For example, the encoding of the JSON object (step 2), can be varied with a different format other than they used in the case implementation example. In the case of the diffusion step, it also can be further adjusted. With regards to the chaos-based system to be used (step 4), it is possible to use other than the one described in Section 4.

Although the implementation example of the proposed approach is enough secure, it can be further improved. It is worth noting that securing information through synchronization of a chaos-based system relies on the concept of super-key, which encompasses the sharing of the same system parameters between the master and slave schemes. If the system parameters are fixed, it can be possible to carry out a synchronization attack, thus gaining access to the system parameters. Once known the system parameters, it is possible to conduct the synchronization between the master and a fake slave scheme, thus having access to the information-bearing signal. A way of mitigating this issue is by strengthening the information-bearing. In our case, we propose to first encode the JSON object and apply a diffusion mechanism over it. An extra security mechanism consists in generating a random noise signal in which the diffused JSON object is merged on it. An extra step is to encode the information-bearing variable along with the driver variable (step 10, see Fig. 1). Complementary, the system parameters of the chaotic system can be dynamically varied, as stated in [37].

Nowadays, JSON is a popular format for interchanging information. In this work, we have proposed an alternative approach for securing JSON objects through the use of chaos-based synchronization. Our approach considers a series of steps that ensure a proper diffusion and confusion of the plaintext to be transmitted. These steps are enough flexible that facilitate adjustments in its implementation.

***Acknowledgments*:** The present work was possible due to collaboration between research group GrIISoft from Escuela Superior Politecnica de Chimborazo, and research group UDG-CA-729 from the CUCIenega – University of Guadalajara. This collaboration is endorsed in document number 043-ESPOCH-2021 granted by the Escuela Superior Politecnica de Chimborazo (Ecuador).

References

1. ECMA. ECMA-404 – The JSON Data Interchange Syntax. Ecma International, 2017 (Online). **https://www.ecma-international.org/publications-and-standards/standards/ecma-404/**
2. B r a y, T. The JavaScript Object Notation (JSON) Data Interchange Format. RFC Editor, 2017 (Online). **https://rfc-editor.org/rfc/rfc8259.txt**
3. B o u r h i s, P., J. L. R e u t t e r, F. S u á r e z, D. V r g o c. JSON: Data Model, Query Languages and Schema Specification. – In: Proc. of 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, May 2017, pp. 123-135. DOI: 10.1145/3034786.3056120.
4. R e s c o r l a, E. The Transport Layer Security (TLS) Protocol Version 1.3. RFC Editor, 2018 (Online). **https://rfc-editor.org/rfc/rfc8446.txt**
5. J o n e s, M., J. H i l d e b r a n d. JSON Web Encryption (JWE). RFC Editor, 2015 (Online). **https://rfc-editor.org/rfc/rfc7516.txt**
6. R a s h i d, F. Y. Critical Flaw Alert! Stop Using JSON Encryption. InfoWorld, 27 March 2017 (Online). **https://www.infoworld.com/article/3184582/critical-flaw-alert-stop-using-json-encryption.html**
7. D e t e r i n g, D., J. S o m o r o v s k y, C. M a i n k a, V. M l a d e n o v, J. S c h w e n k. On The (In-) Security of JavaScript Object Signing and Encryption. – In: Proc. of 1st Reversing and Offensive-Oriented Trends Symposium (ROOTS'17), November 2017, pp. 1-11. DOI:10.1145/3150376.3150379.
8. B l e i c h e n b a c h e r, D. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1," in Advances in Cryptology (CRYPTO'98), 1998, pp. 1-12. DOI: 10.1007/BFb0055716.
9. H u y g e n s, C. Horologium oscillatorium, sive, De motu pendulorum ad horologia aptato demonstrationes geometricae. Paris, F. Muguet, 1673.
10. P e c o r a, L. M., T. L. C a r r o l l. Synchronization in Chaotic Systems. – Phys. Rev. Lett., Vol. **64**, February 1990, No 8, pp. 821-824. DOI: 10.1103/PhysRevLett.64.821.
11. H e, R., P. G. V a i d y a. Analysis and Synthesis of Synchronous Periodic and Chaotic Systems. – Phys. Rev. A, Vol. **46**, December 1992, No 12, pp. 7387-7392. DOI: 10.1103/PhysRevA.46.7387.
12. V a i d y a, P. G. Monitoring and Speeding up Chaotic Synchronization. – Chaos, Solitons & Fractals, Vol. **17**, July 2003, No 2, pp. 433-439. DOI: 10.1016/S0960-0779(02)00384-3.
13. H e, R., P. G. V a i d y a. Implementation of Chaotic Cryptography with Chaotic Synchronization. – Phys. Rev. E, Vol. **57**, February 1998, No 2, pp. 1532-1535. DOI: 10.1103/PhysRevE.57.1532.

14. L o r e n z, E. N. Deterministic Nonperiodic Flow. – J. Atmos. Sci., Vol. **20**, March 1963, No 2, pp. 130-141. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.
15. R ö s s l e r, O. E. An Equation for Continuous Chaos. – Phys. Lett. A, Vol. **57**, July 1976, No 5, pp. 397-398. DOI: 10.1016/0375-9601(76)90101-8.
16. R ö s s l e r, O. E. An Equation for Hyperchaos. – Phys. Lett. A, Vol. **71**, April 1979, No 2, pp. 155-157. DOI: 10.1016/0375-9601(79)90150-6.
17. O p p e n h e i m, A. V., G. W. W o r n e l l, S. H. I s a b e l l e, K. M. C u o m o. Signal Processing in the Context of Chaotic Signals. – In: Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP'92), 1992, IEEE, Vol. **4**, March 1992, pp. 117-120. DOI: 10.1109/ICASSP.1992.226472.
18. D e d i e u, H., M. P. K e n n e d y, M. H a s l e r. Chaos Shift Keying: Modulation and Demodulation of a Chaotic Carrier Using Self-Synchronizing Chua's Circuits. – IEEE Trans. Circuits Syst. II Analog Digit. Signal Process., Vol. **40**, October 1993, No 10, pp. 634-642. DOI: 10.1109/82.246164.
19. D m i t r i e v, A. S., A. I. P a n a s, S. O. S t a r k o v. Experiments on Speech and Music Signals Transmission Using Chaos. – Int. J. Bifurc. Chaos, Vol. **5**, March 1995, No 4, pp. 1249-1254. DOI: 10.1142/S0218127495000910.
20. Y a n g, T., L. O. C h u a. Secure Communication via Chaotic Parameter Modulation. – IEEE Trans. Circuits Syst. I Fundam. Theory Appl., Vol. **43**, September 1996, No 9, pp. 817-819. DOI: 10.1109/81.536758.
21. C u o m o, K. M., A. V. O p p e n h e i m. Circuit Implementation of Synchronized Chaos with Applications to Communications. – Phys. Rev. Lett., Vol. **71**, July 1993, No 1, pp. 65-68. DOI: 10.1103/PhysRevLett.71.65.
22. U c h i d a, A. Optical Communication with Chaotic Lasers: Applications of Nonlinear Dynamics and Synchronization. Wiley, February 2012.
23. Al B a y a t i, B. M., A. K. A h m a d, K. A. M. Al N a i m e e. Effect of Control Parameters on Chaos Synchronization by Means of Optical Feedback. – Opt. Commun., Vol. **472**, April 2020, p. 126032. DOI: 10.1016/j.optcom.2020.126032.
24. T a n g, Y., Q. L i, W. D o n g, M. H u, R. Z e n g. Optical Chaotic Communication Using Correlation Demodulation between Two Synchronized Chaos Lasers. – Opt. Commun., Vol. **498**, November 2021, p. 127232. DOI: 10.1016/j.optcom.2021.127232.
25. M e r a h, L., A. A d n a n e, A. A l i-P a c h a, S. R a m d a n i, N. H a d j-s a i d. Real-Time Implementation of a Chaos Based Cryptosystem on Low-Cost Hardware. – Iran. J. Sci. Technol. Trans. Electr. Eng., Vol. **45**, November 2021, No 4, pp. 1127-1150. DOI: 10.1007/s40998-021-00433-w.
26. L i a o, T.-L., H.-R. L i n, P.-Y. W a n, J.-J. Y a n. Improved Attribute-Based Encryption Using Chaos Synchronization and Its Application to MQTT Security. – Appl. Sci., Vol. **9**, November 2019, No 20, p. 4454. DOI: 10.3390/app9204454.
27. A l m a l i, M. N., Z. D i k i c i. The Simulation of Sound Signal Masking with Different Chaotic Oscillations and Its Circuit Application. – Turkish J. Electr. Eng. Comput. Sci., Vol. **24**, June 2016, pp. 4284-4293. DOI:10.3906/elk-1504-264.
28. V a i d y a n a t h a n, S., A. S a m b a s, S. K a c a r, U. C a v u s o g l u. A New Three-Dimensional Chaotic System with a Cloud-Shaped Curve of Equilibrium Points, Its Circuit Implementation and Sound Encryption. – Int. J. Model. Identif. Control, Vol. **30**, October 2018, No 3, pp. 184-196. DOI: 10.1504/IJMIC.2018.095334.
29. Z a h e r, A. A., G. A m j a d H u s s a i n. Chaos-Based Cryptography for Transmitting Multimedia Data over Public Channels. – In: Proc. of 7th International Conference on Information and Communication Technology (ICoICT'19), July 2019, pp. 1-6. DOI: 10.1109/ICoICT.2019.8835351.
30. Z o u, C., Q. Z h a n g, X. W e i, C. L i u. Encryption Based on Improved Lorenz System. – IEEE Access, Vol. **8**, April 2020, pp. 75728-75740. DOI: 10.1109/ACCESS.2020.2988880.
31. M o o n, S., J.-J. B a i k, J. M. S e o. Chaos Synchronization in Generalized Lorenz Systems and an Application to Image Encryption. – Commun. Nonlinear Sci. Numer. Simul., January 2021. DOI: 10.1016/j.cnsns.2021.105708.
32. P r a s a d, B., K. M i s h r a. A Combined Encryption Compression Scheme Using Chaotic Maps. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 2, pp. 75-81.

33. H u a n g, Q., L. W a n g, G. L i. Research and Application of Video Encryption Technology Based on Chaotic Synchronization Theory. – In: Proc. of 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA'18), April 2018, pp. 444-447. DOI: 10.1109/ICMTMA.2018.00114.
34. J i m é n e z-R o d r í g u e z, M., M. G. G o n z á l e z-N o v o a, J. C. E s t r a d a-G u t i é r r e z, C. A c o s t a-L ú a, O. F l o r e s-S i o r d i a. Secure Point-to-Point Communication Using Chaos. – DYNA, Vol. **83**, Jun. 2016, No 197, pp. 181-187. DOI: 10.15446/dyna.v83n197.53506.
35. R u n g e, C. Über die numerische auflösung von differentialgleichungen. – Math. Ann., Springer, Vol. **46**, June 1895, pp. 167-178. DOI:10.1007/BF01446807.
36. K u t t a, W. Beitrag zur näherungsweisen integration totaler differentialgleichungen. – Zeit. Math. Phy, Vol. **46**, 1901, pp. 435-453.
37. P i s a r c h i k, A. N., M. J i m é n e z-R o d r í g u e z, R. J a i m e s-R e á t e g u i. How to Resist Synchronization Attacks. – Discontinuity, Nonlinearity and Complexity, Vol. **4**, April 2015, No 1, pp. 1-9. DOI: 10.5890/DNC.2015.03.00.

Received: 29.04.2022; Accepted: 16.08.2022