DOI: http://dx.doi.org/10.15446/ing.investig.v38n2.64509

Steganography applied in the origin claim of pictures captured by drones based on chaos

Esteganografía basada en caos, aplicada en la reclamación de origen en imágenes capturadas por drones

Maricela Jiménez-Rodríguez[1], Carlos E. Padilla-Leyferman[2], Juan C. Estrada-Gutiérrez[3], María G. González-Novoa[4], Horacio Gómez-Rodríguez[5], and Octavio Flores-Siordia[6]

ABSTRACT

In this work, steganography is implemented in photographs captured by an unmanned aerial vehicle (drone), with the purpose of adding an identifier that indicates which device they are taken from so it works for the recovery of the origin. In the system, a new technique that modifies the least significant bit (LSB) is applied, using a mathematical model to generate the chaotic orbits, one of the parts selects the RGB channel (Red, Green or Blue) where the LSB is changed and the other is implemented to calculate the random position of the sub pixel to be modified in the selected channel. In addition, a comparison between the bit to be hidden and the LSB of the pixel of the image is performed to verify if it is not necessary to modify it, which lessens the alterations in the container image. It is a tool to capture photos remotely with the Ar.Drone 2.0, with the features needed to perform an analysis that uses correlation diagrams and histograms to verify if the integrity of the message is guaranteed or if changes in the stego-image are visible to the naked eye. On the other hand, a test was done on the Baboon image to compare the robustness of the proposed system with other investigations, evaluating the correlation, contrast, energy, homogeneity, MSE, PSNR and quality index. The results generated were compared with the work of other authors concluding our system provides greater security, integrity, high sensitivity to the keys, it is not linked to a single chaotic system and can be applied to hide imperceptibly all kinds of information, in: radiographs, videos, files, official documents, and other types of containers.

Keywords: Steganography, chaos, security, drones.

RESUMEN

En este trabajo se implementa la esteganografía en las fotografías capturadas por un vehículo aéreo no tripulado (dron), con el fin de agregar un identificador que indique desde qué dispositivo son tomadas, y, de este modo, funcione para la reclamación de origen. En el sistema se aplica una nueva técnica que modifica solo el bit menos significativo (LSB), usando un modelo matemático que se utiliza para generar dos órbitas caóticas, una de las cuales se usa para seleccionar el canal RGB (Rojo, Verde o Azul), donde se cambia el LSB y la otra se implementa para calcular de forma aleatoria la posición del subpíxel que se modifica en el canal seleccionado; además se realiza una comparación entre el bit que se desea ocultar y el LSB del pixel de la imagen, para verificar si no es necesario modificarlo, lo cual altera menos la imagen contenedora. Se realizaron pruebas del algoritmo capturando fotografías de manera remota con un Ar.drone 2.0, con las cuales se hizo un análisis empleando diagramas de correlación e histogramas para verificar si se garantiza la integridad del mensaje o si a simple vista se detectan cambios en el esteganograma. Por otro lado, se realizaron tests a la imagen del Baboon para comparar la robustez del sistema propuesto con los de otras investigaciones, evaluando la correlación, contraste, energía, homogeneidad, MSE, PSNR e índice de calidad. Los resultados generados se compararon con los de otros autores determinando que nuestro sistema proporciona mayor seguridad, integridad, alta sensibilidad a las llaves, además, no está ligado a un solo sistema caótico y se puede aplicar para ocultar de forma imperceptible todo tipo de información, en: radiografías, videos, archivos, documentos oficiales, o cualquier otro tipo de contendor.

Palabras Clave: Esteganografía, caos, seguridad, drones.

Received: April 24th 2017
Accepted: April 25th 2018

[1] Ph.D. in Science and Technology. Affiliation: Professor, Department of Technological Sciences, Universidad de Guadalajara - Mexico. E-mail: maricela.jimenez@cuci.udg.mx.

[2] Computer Systems Engineer: Affiliation: Masters Student, Universidad de Guadalajara, Mexico. E-mail: carlos.pleyferman@alumnos.udg.mx.

[3] Ph.D. in Science. Affiliation: Professor, Universidad de Guadalajara, Mexico. E-mail: jcarlos.estrada@cuci.udg.mx.

[4] M.Sc. in Applied Computing. Affiliation: Professor, Universidad de Guadalajara, Mexico. E-mail: gleznogpe@hotmail.com.

[5] M.Sc. in Applied Computing. Affiliation: Professor, Universidad de Guadalajara, Mexico. E-mail: horacio.gomez@cualtos.udg.mx.

[6] Ph.D. in Teaching Methodology. Affiliation: Professor, Universidad de Guadalajara, Mexico. E-mail: o_flores@live.com.mx.

How to cite: Jiménez-Rodríguez, M., Padilla-Leyferman, C. E., Estrada-Gutiérrez, J. C., González-Novoa, M. G., Gómez-Rodríguez, H., Flores-Siordia, O. (2018). Steganography applied in the origin claim of pictures captured by drones based on chaos. *Ingeniería e Investigación*, 38(2), 61-69. DOI: 10.15446/ing.investig.v38n2.64509

Introduction

Currently, drones have acquired great importance, since they allow us to reach difficult-to-access areas remotely. Therefore, the Mexican Ministry of Communications and Transportation has indicated they should only be employed during daytime hours in areas not classified as prohibited, restricted, or dangerous (SCT, 2015). Drones are an excellent tool used in security matters regarding different types of transport (Reshma, Ramesh, & Kumar, 2015). However, drones possess vulnerabilities, such as those found in a security analysis where it was determined that an attacker can have access to them and obtain information due to the management of control commands, telemetry, data, and video via Wireless Local Area Networks (WLAN) without any type of encryption (Samland, Fruth, Hildebrandt, & Dittman, 2012). Thus, it is necessary to implement security measures, with the cryptography focused in the encryption of data not to be interpreted, or with the steganography specialized in data concealment (Morocho-Checa, Zambrano-Miranda, Carvajal-Rodríguez, & Lopez-Fonseca, 2015). There are systems that take advantage of the information technologies to apply steganography in computing systems (Morocho-Checa, Zambrano-Miranda, Carvajal-Rodríguez, & Lopez-Fonseca, 2015; Rodríguez-Mendoza, 2016). In addition, secret messages can be embedded inside different multimedia containers, exploiting their intrinsic properties, such as spaces in text documents, pixels in images, and frames in videos (Méndez-Naranjo, 2015; Morocho-Checa, Zambrano-Miranda, Carvajal-Rodríguez, & Lopez-Fonseca, 2015; Rodríguez-Mendoza, 2016). Steganography was applied to cloak a message inside a picture, where the pixels are modified according to the Least Significant Bit (LSB) placed along the edges of the image (Ratnakirti, Anirban, & Suvamoy, 2013). Anees et al. proposed a technique to apply digital steganography in which the authors replaced two different areas of an image according to LSB and Most Significant Bits (MSB) (Anees, Siddiqui, Ahmed, & Hussain, 2014). Ranjith-Kumar et al. (2016) proposed a technique to modify LSB and Intermediate Significant Bits (ISB), which enhances the quantity of positions that can be altered by applying steganography; in addition, these authors implemented the technique of confusion in the plain text prior to inserting it into the container. Martínez-González et al. designed a system where they take the plain text and divide it into four bit sections for insertion into the four LSB of an image (Martínez-González, Díaz-Méndez, Palacio-Luengas, López-Hernández, & Vázquez-Medina, 2016). Methods for encoding data have been developed through implementing chaotic systems, due to their desired properties, such as sensitivity to the initial conditions and parameters, rendering them efficient for the design of security applications due to their versatility and complex evolution. In telecommunications, packages with chaotic encryption of poor analytical rigor were sent with satisfying results (Jiménez-Rodríguez, González-Novoa, Estrada-Gutiérrez, Acosta-Lua, & Flores-Siordia, 2016). A security and authentication scheme was developed via a secret 128-bit key, employing a tent map (Sen-Te & Samsudin, 2017). Kanso and Ghebleh (2017) implemented an algorithm with the Arnold's cat map, altering the frequency of the image instead of its pixels; the changes are notable in color images, limiting its use for blacks and whites. A transmission method for combining ElectroCardioGraphy (ECG) signals obtained from a patient with chaotic algorithms was proposed (Bárbara-Morales, Alba-Blanco, & Rodríguez-Ramírez, 2012). Nassar et al. (2016) implemented a scheme where encryption and steganography are applied, utilizing the Logistic map for encryption and noise conversion and the Baker map for steganography. Sharif et al., designed a system in which chaotic maps are implemented to generate three positions plus dividing the byte-to-insert into four bits, which are embedded in the container, replacing the four LSB of two positions (Sharif, Mollaeefar, & Nazari, 2016). A scheme consisting of the insertion of a message into a container was proposed; the resulting stego-image is encrypted and inserted into another container afterward (Baig, Khan, Beg, Shah, & Saleem, 2016). Xiang et al., developed an algorithm in which each character of the message is divided into two, four-bit sections, where the MSB and LSB are placed in two positions generated by the chaotic system, replacing its four LSB. This algorithm is designed to work with black-and-white pictures and a message shorter in size than that of the container image (Xiang, Hu, & Sun, 2015). Ghebleh and Kanso apply criteria to generate values between 5 and 250 and to focus on edge detection within a container image, where they hide information in the frequency domain (Ghebleh & Kanso, 2014). Aziz et al., designed a system consisting of two phases: in the first, the message-to-conceal is encrypted as noise, and in the second, the resulting cryptogram is embedded in the container image. The selected byte is divided into three-bit segments to be inserted into the LSB of a pixel. This is designed to work with images of 256 colors or grayscales (Aziz, Tayarami-N, & Afsar, 2015).

Drones are widely used, because they allow video recording and taking pictures of places where access is difficult for persons. However, there can be malicious users remotely connected to the drone who intend to take possession of the photos or videos recorded by it. Thus, it is necessary to implement measurements that permit origin claim; in this manner, it could be determined from which device the images were captured. With that goal in mind, this paper implemented a system connected to a drone and applying a novel steganography technique to identify the device employing chaos. It is focused on the LSB scheme, but only modifies it a little, differentiating it from systems in which three or four are replaced (Aziz, Tayarami-N, & Afsar, 2015; Martínez-González, Díaz-Méndez, Palacio-Luengas, López-Hernández, & Vázquez-Medina, 2016; Xiang, Hu, & Sun, 2015). The aim of this research is the development of a system capable of hiding an identifier in a manner imperceptible to the human eye in the photographs captured from a drone, to safeguard the copyright of the owner of the device, implementing a steganography technique that selects in chaotic way the RGB colors of the image and modifies only the least significant bit of the selected channel. The algorithm was developed in Java, so it is completely portable and can be executed in multiple platforms of operating systems or in single board computers that adapt to the drones with insufficient processing capacity. The steganography method can be used in different research areas to hide information in containers such as: images, videos, bio-signals, data and audio, among others. The remainder of this article is organized as follows: in the methodology explaining the process to apply steganography in images is displayed, in discussion and results analysis section, made different tests to verify the security of the system is exhibited, in conclusions we obtained are detailed.

Methodology

For follow-up, the mathematical model used for the system tests is explained. It is important to highlight that any other chaotic model can be implemented.

Chaotic system: Rössler oscillator

In 1976, Otto E. Rössler found a relatively simple system that possesses an elemental geometric building of chaos in continuous systems and that is represented by the set of following Equations (1a) to (1c).

$$\frac{dx}{dt} = -(y + z) \tag{1a}$$

$$\frac{dy}{dt} = x + ay \tag{1b}$$

$$\frac{dz}{dt} = b + z(x - c) \tag{1c}$$

The equation system is composed of three initial conditions (x_0,y_0,z_0) and three control parameters (a,b,c), which are utilized as encryption keys (Rössler, 1976).

System operation

The system can hide any type of data inside images that are composed for pixels, and these, in turn for subpixels (*RGB*), which have values between 0 and 255.

To conceal information, the Rössler oscillator is employed to generate three chaotic orbits, according to their variables (x,y,z), from which one is utilized to select the pixel of the image, another, the subpixel (*RGB*) and, in case of applying steganography in video, the latter is used to choose a random photogram.

Here below, the steps are explained for using the Rössler oscillator:

1. Capture the image from the drone, which will serve as our container file to hide its identifier.
2. Store the identification in a file.
3. Convert the identification into binary code.
4. Implement the proposed algorithm to conceal the identification in the container file.
5. Create the stego-image.

This process is illustrated in Figure 1.

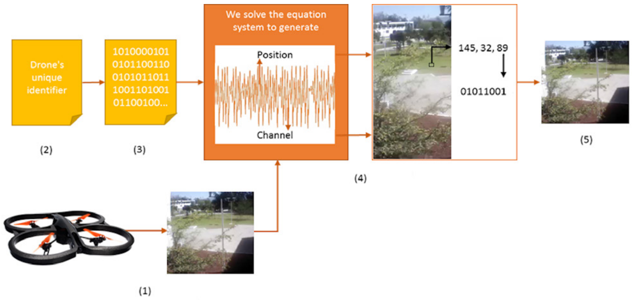


Figure 1. System operation flow chart.
Source: Authors

Following this, the process to embed the identification inside an image is explained, although any type of information in bit format can be inserted as well.

Nomenclature

IC: Original or container image where the message will be randomly hidden.

M: Message or identifier to be sent as concealed within the container.

Ste: Image after the message has been inserted (stego-image).

Encryption keys:

x_0,y_0,z_0: values assigned to the initial conditions.

a,b,c: values of the control parameters.

Algorithm 1: Stego-image generator

Step 1. Store in variable *LM* the characters contained in message *M*.

Step 2. Convert *LM* characters, blank spaces, symbols, letters, numbers, into their respective binary value (0 and 1) and store them in *MB*.

$$MB = 0110100101010110011001011$$

Step 3. Determine the length of *MB*, which would be equal to multiplying the length of the *LM* message times

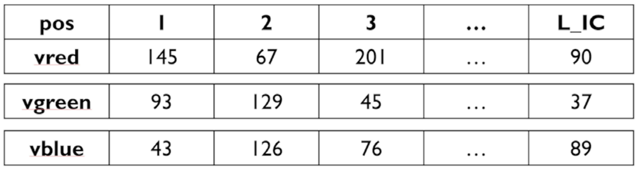
8 (corresponding to the number of bits for each character) and assign it to *LMB*.

$$LMB = LM * 8$$

Step 4. Calculate the number of pixels stored in image *IC*, to determine the length of the vectors in which they will be saved afterward (multiplying the width by height of the image).

$$L\_IC = WidthIC * HeightIC$$

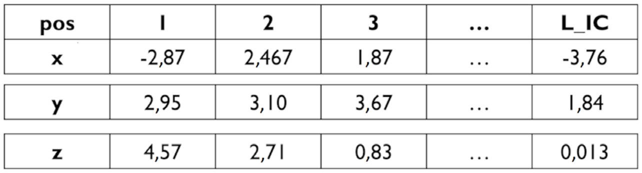
Step 5. Assign the values of each corresponding *RGB* pixel of image *IC*, in their respective vector.



pos	1	2	3	...	L_IC
vred	145	67	201	...	90
vgreen	93	129	45	...	37
vblue	43	126	76	...	89

Figure 2. Vectors with color RGB.
Source: Authors

Step 6. Solve the system of Equations (1a) to (1c) with the keys (initial conditions and parameters), to generate 3 vectors where the chaotic orbits *x*, *y*, *z*, are stored, this step is repeated until a vector of length *L_IC* is generated.



pos	1	2	3	...	L_IC
x	-2,87	2,467	1,87	...	-3,76
y	2,95	3,10	3,67	...	1,84
z	4,57	2,71	0,83	...	0,013

Figure 3. Vectors *x*, *y*, *z* with chaotic orbits.
Source: Authors

Step 7. Take a term of *x* to divide it between the absolute value of the largest number of the vector *x*, in order to generate a float between 0 and 1. The result is stored variable *norm*.

$$norm = \frac{x[pos]}{|\max\_val\_vector(x)|}$$

Step 8. Calculate location *loc* multiplying *L_IC* by norm. Afterward, the result is rounded off and its absolute value is taken. Therefore, a position between 0 and *L_IC* will be generated.

$$loc = |round\ (L\_IC * norm)|$$

Step 9. Determine the *RGB* color channel to modify, taking a value of *y*, generated in step 6. It is divided by the largest number of its vector and multiplied by 2; afterward, it is rounded off and the absolute value, obtaining a number indicating the *RGB* vector that will be altered, which can be one of the following: 0 modifies vred, 1 alters vgreen or 2 vblue.

$$color = \left|round\left(\frac{y[pos]}{\max\_val\_vector(y)} * 2\right)\right|$$

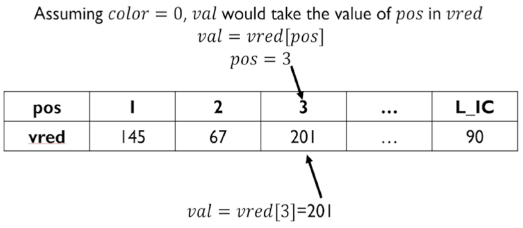
Step 10. Verify whether *color* equals:

a. 0, take value *val* stored in position *pos* of the array vred[pos].

b. 1, take value *val* stored in position *pos* of the array vgreen[pos].

c. 2, take value *val* stored in position *pos* of the array vblue [pos].

d. As shown in the example in Figure 4.



Assuming $color = 0$, val would take the value of pos in $vred$
$val = vred[pos]$
$pos = 3$

pos	1	2	3	...	L_IC
vred	145	67	201	...	90

$val = vred[3]$=201

Figure 4. The process to select the vector.
Source: Authors

Step 11. Verify that the array and position selected in Step 10 have not been modified previously; otherwise, repeat Steps 7–10.

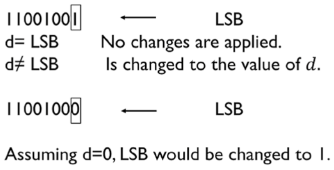
Step 12. Modify the Least Significant Bit (LSB).

- Convert *val* to binary: *val* = 201 = 11001001.
- Take binary digit *d* from *MB* and verify.

a. If d equals the *LSB* of *val*, it remains as it is;

b. Otherwise, change *LSB* (if its 0 is replaced by 1; on the other hand, if it is 1, it is changed to 0).

As shown in the example in Figure 5.



11001001 ⟵ LSB
d= LSB No changes are applied.
d≠ LSB Is changed to the value of *d*.

11001000 ⟵ LSB

Assuming d=0, LSB would be changed to 1.

Figure 5. Evaluation to determine whether the Least Significant Bit (LSB) requires modification.
Source: Authors

Step 13. Convert *val* back to decimal and store it in the vector position from which it was taken in Step 10.

Step 14. Repeat Steps 7 to 13 until *MB* has been concealed in its whole length *IC*.

On concluding the previously mentioned 14 Steps, stego-image *Ste* is generated.

Algorithm 2: Recovering the information hidden in the stego-image

For recovery, it is necessary to have stego-image **Ste**, the cipher keys (parameters and initial conditions used in Step 6 of Algorithm 1), and length of message **LMB** generated in Step 3 of Algorithm 1.

Step 1. Calculate the number of pixels stored in stego-image *Ste*.

$$L\_IC = WidthSte * HeightSte$$

Step 2. Repeat Steps 5–11 of Algorithm 1.

Step 3. Convert val to binary: *val* = 200 = 11001000.

Take the *LSB* of val and concatenate to the end of string *MB*, as shown in Figure 6.

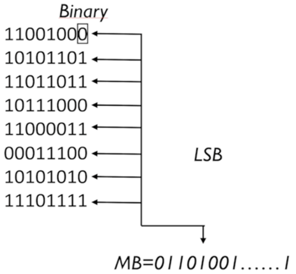


Figure 6. Recover the Least Significant Bits (LSB).
Source: Authors

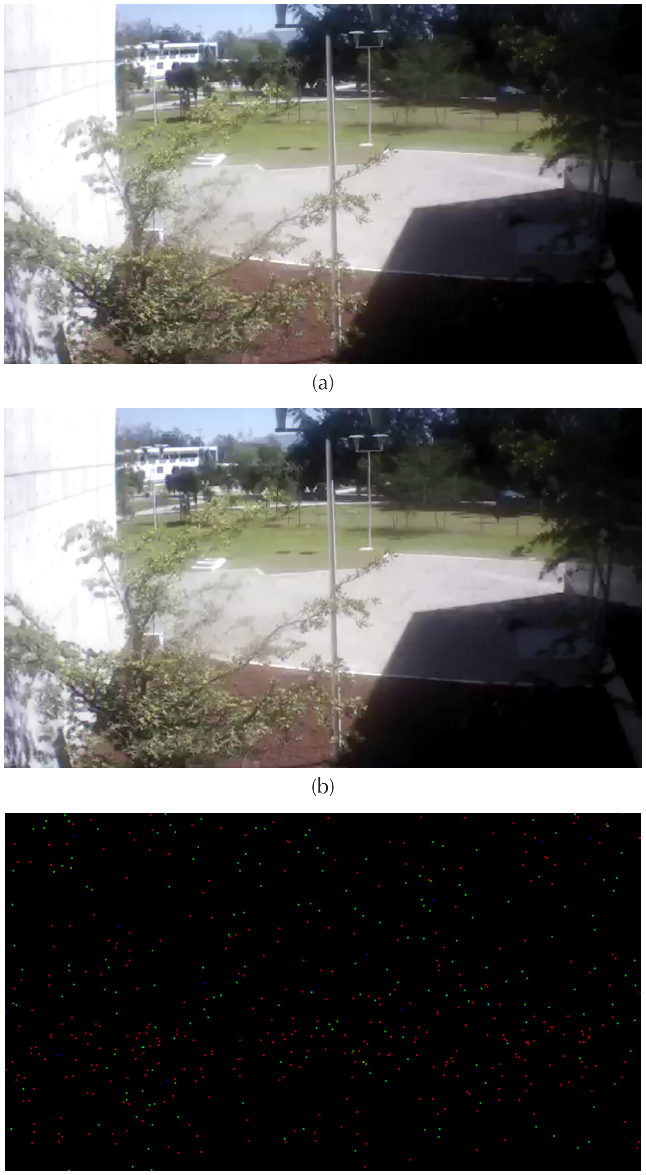
$$MB = \texttt{0110100101010110011001011}$$

Step 4. Repeat Steps 2 and 3 until MB is of a length equal to LMB.

Step 5. Group *MB* in blocks of 8 bits. Each byte is then converted into its respective ASCII value; thus, hidden message *M* is recovered.

Discussion and Results Analysis

The algorithm was implemented in Java, a multiplatform programming language that allows its execution in any system compatible with the drone or in mobile or desktop applications. We promptly displayed the results obtained in the system from a picture captured by an AR.Drone 2.0, where 1 819 characters were hidden. In Figure 7(a), the original image is displayed, the stego-image is exhibited in Figure 7(b) and, as can be observed, the subpixels in three colors of the stego-image in Figure 7(b) were modified, this imperceptible to the human eye.



(a)

(b)

c)

Figure 7. (a) Original image, (b) Stego-image, and (c) Modified pixels in Stego-mage.
Source: Authors

In Figure 7(c), the pixels were the LSB in the modified stego-image are shown, marked with red, green, and blue dots depending on the altered RGB channel. The remainder are displayed in black.

Speed

Tests were performed using the system to conceal messages of different sizes within a 640 × 360 pixel image, captured by the AR.Drone 2.0. The results are portrayed in Table 1. In the column size of the message, the number of characters is indicated that are hidden in the stego-image. In the second, the time taken by the algorithm to hide the data is exhibited, while in the third, the time it took to recover the message from the stego-image is displayed. In the very last, we find the percentage of subpixels altered in the container image.

Table 1. Speed of the system applying steganography

Message size	Concealment time (ms)	Recovery time (ms)	Subpixels
1 819	278	245	2,10%
1 045	240	140	1,21%
531	222	110	0,61%
209	220	72	0,24%

Source: Authors

Statistical analysis

Correlation analysis

To verify the difference between pixels of the container image and of the stego-image, the diagram exhibited in Figure 8 was developed. In this diagram, correlation coefficient value R = 1 can be observed; therefore, the alterations of the stego-image are practically imperceptible, given that the modifications were quite small due to changing only one bit.

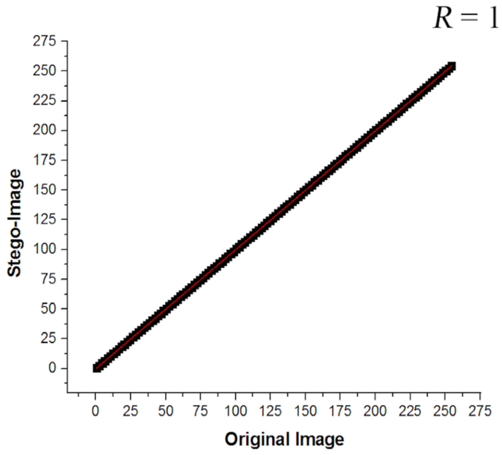


Figure 8. Correlation diagram: original image vs stego-image.
Source: Authors

Another correlation analysis was developed comparing the data recovered from stego-image with the original message hidden within it, to verify whether the data was recovered completely or whether there was some loss in the tracing. The diagram is shown in Figure 9. The resulting correlation coefficient is R = 1; therefore, there was no data loss at the time that the message was recovered.

The correlation diagram of Figure 10 was performed to verify the sensitivity of the keys used in the algorithm by the Rössler oscillator to hide the information. In the diagram, the relation between the original message hidden with key values: x_0 = 3,451458, y_0 = 1,952658, z_0 = 7,871425, a = 0,2, b = 0,2, c = 9,5 and the recovered message with the following values: x_0 = 3,451458, y_0 = 1,952658, z_0 = 7,871425, a = 0,1, b = 0,1, c = 9,2 as displayed in the correlation coefficient R = -0,00546, indicating that the relation between the hidden message and the recovered one is practically null, this sustaining the security and confidentiality of the information. Thus, if an attacker intercepts the stego-image, the correct keys are necessary to recover the message.

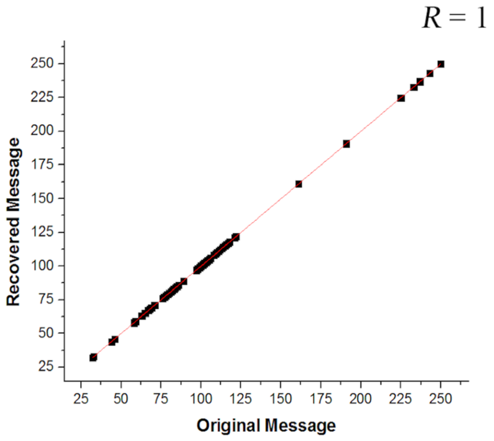


Figure 9. Correlation diagram: original message vs recovered message.
Source: Authors

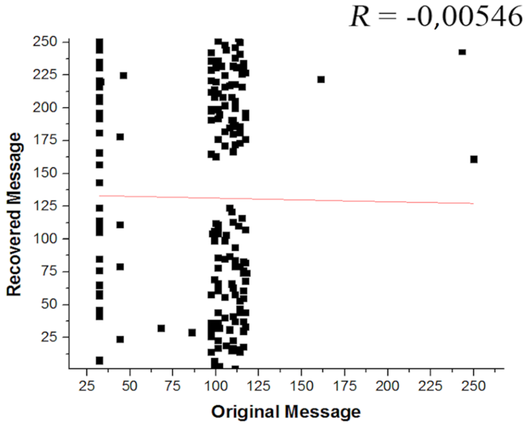


Figure 10. Correlation diagram: original message vs recovered message with different keys.
Source: Authors

Histograms

In Figure 11, histograms graphically representing the pixel distribution of the original image and of the stego-image are displayed.

Figures 11(a), (c) and (e) present the histograms corresponding to the original image of red, green and blue colors, respectively. In Figures 11(b), (d) and (f), the histograms of the stego-image are exhibited.

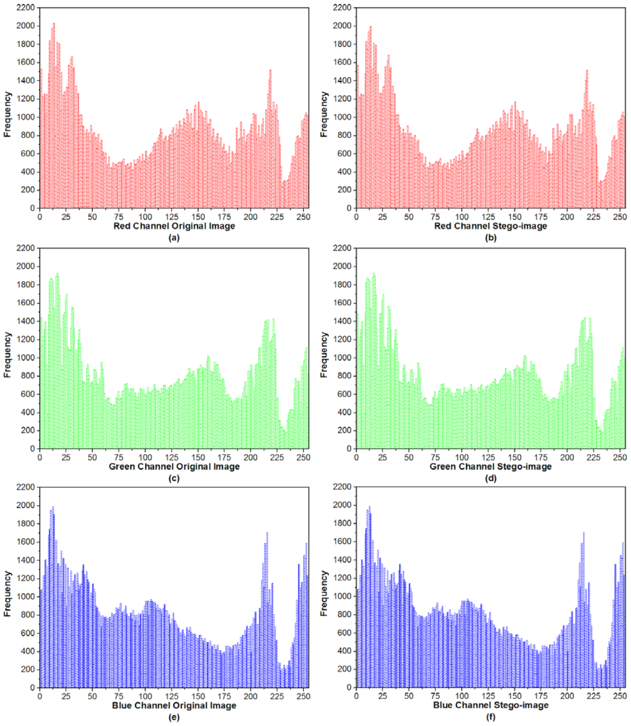


Figure 11. Histograms of the original image. (a), (c), (e), red, green and blue subpixels, respectively. Histograms of the stego-image (b) red, (d) green, and (f) blue.
Source: Authors

Comparing the histogram color pairs, significant changes are not noticeable, this due to the altering of a single bit (the LSB). Ergo, subpixel values in positions randomly chosen by the orbit are only added to or subtracted from 1, but only if this is necessary, therefore rendering the alterations of the image very slight.

Stego-analysis

To detect the small differences between the steganogram and the original image, the statistical security analyzes shown by Anees, Siddiqui, Ahmed, & Hussain (2014) is performed to determine the correlation (Corr), contrast (C), energy (E) and homogeneity (Hom). The correlation of an image is given by:

$$Corr = \sum_{i,j} \frac{(i-\mu i)(j-\mu j)\, p(i,j)}{\sigma j \sigma j} \tag{2}$$

The contrast of an image is given by:

$$C = \sum_{i,j} |i-j|^2 \, p(i,j) \tag{3}$$

The energy of a digital image defined as:

$$E = \sum_{i,j} p(i,j)^2 \tag{4}$$

The homogeneity analysis processes can be determined by:

$$Hom = \sum_{i,j} \frac{p(i,j)}{1-|i-j|} \tag{5}$$

To compare the proposed system with other authors, a message is hidden in the image of Baboom Figure 12 (a) and a steganogram is generated Figure 12 (b).

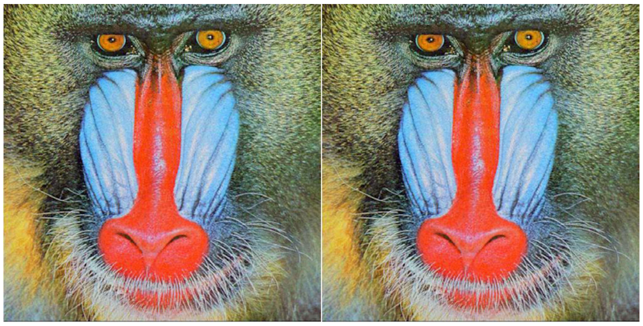


Figure 12. (a) Original image, (b) Stego-image.
Source: Original Baboon image

Table 2 shows the results of the statistical analysis applied to Figures 12 (a) and 12 (b), and they are compared with the data of other authors in which they apply the methods Equations (2) to (5) to the same image, in the results it is observed that the technique proposed by Sharif, Mollaeefar, & Nazari (2016) like ours does not present alterations between the original image and the steganogram. Unlike the slight changes generated by the method of Anees, Siddiqui, Ahmed, & Hussain (2014).

Table 2. Comparative statistical analysis

Analysis	Original			Steganogram			Method
	Red	Green	Blue	Red	Green	Blue	
Contrast	1,4706	1,5046	1,5263	1,4706	1,5046	1,5263	Proposed
Correlation	0,7646	0,6662	0,7971	0,7646	0,6662	0,7971	
Energy	0,0488	0,0537	0,0443	0,0488	0,0537	0,0443	
Hom	0,6983	0,6882	0,6847	0,6983	0,6882	0,6847	
Contrast	0,6420	0,6449	0,6598	0,6421	0,6449	0,6599	(Anees, Siddiqui, Ahmed, & Hussain, 2014)
Correlation	0,8582	0,7827	0,8698	0,8582	0,7827	0,8698	
Energy	0,0760	0,0958	0,0771	0,0760	0,0958	0,0771	
Hom	0,7781	0,7817	0,7754	0,7781	0,7817	0,7754	
Contrast	0,7425	0,7847	0,7546	0,7425	0,7847	0,7546	(Sharif, Mollaee-far, & Nazari, 2016)
Correlation	0,8829	0,8205	0,8929	0,8829	0,8205	0,8929	
Energy	0,0456	0,0463	0,0463	0,0456	0,0463	0,0463	
Hom	0,6970	0,6769	0,6774	0,6970	0,6769	0,6774	

Source: Authors

Other types of analysis are implemented, such as MSE, PSNR and SSIM (Index quality).

Where Mean Squared Error (MSE), it can be defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(X_i - X_i^*\right)^2 \tag{6}$$

The Pick Signal to Noise Ratio (PSNR), it is given as:

$$PSNR = 10 \bullet \log_{10}\left(\frac{MAX_1^2}{MSE}\right) \tag{7}$$

The Structural Similarity (SSIM) quality Index is defined as:

$$SSIM(x, y) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{8}$$

Table 3 presents the MSE values calculated by Equation (6) and PSNR with Equation (7) of Figures 12 (a) and 12 (b), the results are compared with Anees, Siddiqui, Ahmed, & Hussain (2014). There, it is observed that our method presents a higher PSNR, indicating better quality in the image, on the other hand the MSE is lower therefore few differences were detected.

Table 3. Comparison of MSE and PSNR according to the RGB colors

Frame	Proposed		(Anees, Siddiqui, Ahmed, & Hussain, 2014)	
	MSE	PSNR	MSE	PSNR
Red	0,0046	71,5377	0,0441	55,4126
Green	0,0023	74,4687	0,0108	55,4126
Blue	0,0002	84,9838	0,0025	55,4126

Source: Authors

The quality index and PSNR are calculated by Equations (7) and (8) with Baboon images of (256 x 256) and (512 x 512), the results are shown in Tables 4 and 5, in which it can observed that the proposed system displays better results.

Table 4. Comparison of quality analysis and PSNR for Baboon of (256 x 256)

Chars of message	Bits of message	Proposed		(Sharif, Mollaeefar, & Nazari, 2016)	
		PSNR	Quality index	PSNR	Quality index
10 000	80 000	65,5024	0,9999892005544526	43,1189	0,999605430887160
20 000	160 000	69,3302	0,9999955363878346	40,2507	0,999682914275348
30 000	240 000	67,3555	0,9999929432912952	38,7540	0,999652148974817

Source: Authors

The results of the statistical analyzes of contrast, correlation, energy, homogeneity, MSE, PSNR and quality index presented in Tables 2-5 were generated in Matlab software.

Table 5. Comparison of quality analysis and PSNR for Baboon of (512 x 512)

Chars of message	Bits of message	Proposed		(Sharif, Mollaeefar, & Nazari, 2016)	
		PSNR	Quality index	PSNR	Quality index
10 000	80 000	74,3931	0,9999983954375297	48,9988	0,999715799434268
20 000	160 000	71,2811	0,9999967605743416	46,1938	0,999705306617473
30 000	240 000	75,0786	0,9999986332966896	44,6715	0,999695228770336

Source: Authors

Conclusions

The proposed system to conceal information within pictures taken by an unmanned aerial vehicle to identity from which device they were taken was implemented in an AR.Drone 2.0; it should be noted that the latter can be utilized in devices without the necessary resources by adding additional hardware or an embedded system to capture, process, and apply steganography in the image. The system guarantees the reliability of exploiting the properties of chaotic mathematic models, such as high sensitivity to the parameters and to the initial condition, employed as cipher keys to hide the message. Thus, only users with access to these can recover the message. Further, it was designed to use any mathematic model and not a specific one; ergo, it can implement any chaos-generating model with the sole requirement of generating two orbits for applying steganography in images and three for video. Additionally, the system fortifies the security by only altering a single bit at a time within the container (the least significant of these, the LSB) and only when it is different from the bit-to-hide. The previously mentioned information makes the modification of the images lessen, and it is more difficult to perceive the changes between the original image and the stego-image; this was tested in the statistical analysis evaluating frequency, the correlation, contrast, energy, homogeneity, MSE, PSNR, and quality index of the images. The system hides the message in the three RGB channels by inflicting chaos inside the whole image; not just the edges; this guarantees the concealment of bigger data and increases the difficulty for an attacking cryptanalysis. The system assures to the 100% integrity of the hidden message checked through the correlation diagrams, where the original message and the recovered message from the stego-image were compared. Therefore, high sensitivity to the cipher keys was verified with another diagram, evaluating the lineal association between the original message and the recovered message with different keys. The proposed system can be used in different research areas that entail working with images, such as Medicine, Radiology, Photogeology, among others.

Acknowledgements

This work is supported by CONACYT in the project 743187.

References

Anees, A., Siddiqui, A. M., Ahmed, J., & Hussain, I. (2014, March). A technique for digital steganography using chaotic maps. *Nonlinear Dynamics*, 75(4), 807-816. DOI:10.1007/s11071-013-1105-3

Aziz, M., Tayarami-N, M. H., & Afsar, M. (2015, May). A cycling chaos-based cryptic-free algorithm for image steganography. *Nonlinear Dynamics*, 80(3), 1271-1290. DOI:10.1007/s11071-015-1943-2

Baig, F., Khan, M. F., Beg, S., Shah, T., & Saleem, K. (2016, May). Onion steganography: a novel layering approach. *Nonlinear Dynamics*, 84(3), 1431-1446. DOI:10.1007/s11071-015-2580-5

Bárbara-Morales, E., Alba-Blanco, E., & Rodríguez-Ramírez, O. (2012, May). Modulating electrocardiographic signals with chaotic algorithms. *Ingeniería e Investigación*, 32(2), 46-50. Retrieved from https://revistas.unal.edu.co/index.php/ingeinv/article/view/31939/34557.

Ghebleh, M., & Kanso, A. (2014, June). A robust chaotic algorithm for digital image steganography. *Communications in Nonlinear Science and Numerical Simulation*, 19(6), 1898-1907. DOI:10.1016/j.cnsns.2013.10.014

Jiménez-Rodríguez, M., González-Novoa, M. G., Estrada-Gutiérrez, J. C., Acosta-Lua, C., & Flores-Siordia, O. (2016, May). Secure point-to-point communication using chaos. *DYNA*, 83(197), 181-187. DOI:10.15446/dyna.v83n197.53506

Kanso, A., & Ghebleh, M. (2017, March). An algorithm for encryption of secret images into meaningful images. *Optics and Lasers in Engineering*, 90(1), 196-208. DOI:10.1016/j.optlaseng.2016.10.009

Martínez-González, R. F., Díaz-Méndez, J. A., Palacio-Luengas, L., López-Hernández, J., & Vázquez-Medina, R. (2016, August). A steganographic method using Bernoulli's chaotic maps. *Computers and Electrical Engineering*, 54(C), 435-449. DOI:10.1016/j.compeleceng.2015.12.005

Méndez-Naranjo, P. M. (2015). Nuevo Algoritmo Criptográfico con la incorporación de la esteganografía en imágenes. Riombamba: Escuela Superior Politécnica de Chimborazo. Retrieved from http://www.dspace.uce.edu.ec/bitstream/25000/6356/1/T-UCE-0011-261.pdf

Morocho-Checa, E., Zambrano-Miranda, J. A., Carvajal-Rodríguez, J. E., & Lopez-Fonseca, G. R. (2015, April). Análisis del Algoritmo Esteganográfico F5 para Imágenes JPEG a Color. *Revista Politécnica*, 36(3), 79-85. Retrieved from http://www.revistapolitecnica.epn.edu.ec/images/revista/volumen36/tomo3/AnalisisdelAlgoritmoEsteganograficoF5paraImagenesJPEGa.pdf

Nassar, S. S., Ayad, N. M., Kelash, H. M., El-sayed, H. S., El-Bendary, M. A., El-Samie, F. E., & Faragallah, O. S. (2016, December). Secure Wireless Image Communication Using LSB Steganography and Chaotic Baker Ciphering. *Wireless Pers Commun*, 91(3), 1023–1049. DOI:10.1007/s11277-016-3387-5

Ranjith-Kumar, R., Jayasudha, S., & Pradeep, S. (2016, November). Efficient and secure data hiding in encrypted images: A new approach using chaos. *Information Security Journal: A Global Perspective*, 25(46), 235–246. DOI:10.1080/19393555.2016.1248582

Ratnakirti, R., Anirban, S., & Suvamoy, C. (2013, December). Chaos based Edge Adaptive Image Steganography. *Procedia Technology*, 10(1), 138-146. DOI:10.1016/j.protcy.2013.12.346

Reshma, R., Ramesh, T. K., & Kumar, P. S. (2015). Security Incident Management in Ground Transportation System Using UAVs. *6th IEEE International Conference on Computational Intelligence* (pp. 1-7). Madurai: IEEE. DOI:10.1109/ICCIC.2015.7435627

Rodríguez-Mendoza, M. N. (2016, May 18). Análisis de las técnicas de esteganografía para el ocultamiento de información. *Tesis - Ingeniería en Informática*. Quito, Pichincha, Ecuador: Universidad Central de Ecuador. Retrieved from http://www.dspace.uce.edu.ec/bitstream/25000/6356/1/T-UCE-0011-261.pdf

Rössler, O. E. (1976, July). An Equation for Continuous Chaos. *Physical Letters*, 57(5), 397-398. DOI:10.1016/0375-9601(76)90101-8

Samland, F., Fruth, J., Hildebrandt, M., & Dittman, J. (2012). AR.Drone: Security threat analysis and exemplary attack to track persons. *Proceedings of SPIE - The International Society for Optical Engineering* (pp. 1-15). Magdeburg: SPIE. DOI:10.1117/12.902990

SCT, C. S. (2015, April 4). Regula la SCT el uso de Aeronaves No Tripuladas (Drones). Retrieved from Secretaría de Comunicación y Transportes: http://www.sct.gob.mx/despliega-noticias/article/regula-la-sct-el-uso-de-aeronaves-no-tripuladas-drones/

Sen-Te, J., & Samsudin, A. (2017, January). A Chaos-Based Authenticated Cipher with Associated Data. *Security and Communication Networks*, 17(1), 1-15. DOI:10.1155/2017/9040518

Sharif, A., Mollaeefar, M., & Nazari, M. (2016, March). A novel method for digital image steganography based on a new three-dimensional chaotic map. *Multimedia Tools and Applications*, 1-19. DOI:10.1007/s11042-016-3398-y

Xiang, T., Hu, J., & Sun, J. (2015, August). Outsourcing chaotic selective image encryption to the cloud with steganography. *Digital SignalProcessing*, 43(C), 28-37. DOI:10.1016/j.dsp.2015.05.006