# On the security of a new image encryption scheme based on chaotic map lattices[*]

David Arroyo,[1, †] Rhouma Rhouma,[2] Gonzalo Alvarez,[1] Shujun Li,[3] and Veronica Fernandez[1]

*[1] Instituto de Física Aplicada, Consejo Superior de Investigaciones Científicas, Serrano 144—28006 Madrid, Spain*
*[2] Syscom Laboratory, Ecole Nationale d'Ingénieurs de Tunis, 37, Le Belvédère 1002 Tunis, Tunisia*
*[3] FernUniversität in Hagen, Lehrgebiet Informationstechnik, Universitätsstraße 27, 58084 Hagen, Germany[‡]*

This paper reports a detailed cryptanalysis of a recently proposed encryption scheme based on the logistic map [Pisarchik *et* al., Chaos 16:033118, 2006]. Some problems are emphasized concerning the key space definition and the implementation of the cryptosystem using floating-point operations. It is also shown how it is possible to reduce considerably the key space through a ciphertext-only attack. Moreover, a timing attack allows the estimation of part of the key due to the existent relationship between this part of the key and the encryption/decryption time. As a result, the main features of the cryptosystem do not satisfy the demands of secure communications. Some hints are offered to improve the cryptosystem under study according to those requirements.

**Recently a new cryptosystem was proposed by using a chaotic map lattice (CML). In this paper, we analyze the security of this cryptosystem and point out some of its security defects. A number of measures have been suggested to enhance the security of the cryptosystem following some established guidelines on how to design good cryptosystems with chaos.**

## I. INTRODUCTION

Image encryption is somehow different from text encryption due to some inherent features of images, such as bulk data capacity and high correlation among pixels. Therefore, digital chaotic ciphers like those in [1–3] and traditional cryptographic techniques such as DES, IDEA and RSA are no longer suitable for practical image encryption, especially for real-time communication scenarios. So far, many chaos-based image cryptosystems have been proposed [4–8]. The major core of these encryption systems consists of one or several chaotic maps serving the purpose of either just encrypting the image or shuffling the image and subsequently encrypting the resulting shuffled image. In [9] a new image encryption algorithm based on chaotic map lattices has been proposed. The aim of this paper is to assess the security of such cryptosystem.

The rest of the paper is organized as follows. Section II describes the cryptosystem introduced in [9]. After that, Section III points out some design problems inherent to that cryptosystem, and Section IV gives some attacks on the cryptosystem under study. Finally, some security enhancements are presented in Section V followed by the last section, which presents the conclusions.

## II. DESCRIPTION OF THE ENCRYPTION SCHEME

The encryption scheme described in [9] is based on the logistic map given by

$$x_{n+1} = a \cdot x_n \cdot (1 - x_n). \tag{1}$$

For a certain value of $a$, the chaotic phase space is $[x_{\min}, x_{\max}]$.

Given an $M \times N$ color image with R, G, B color components, an initialization process is performed to convert the integer values of each pixel to real numbers that can be encrypted using the above chaotic logistic map. First, the 2-D image is scanned in the raster order (i.e., from left to right and from top to bottom) to form three 1-D integer sequences $\{P_c^i\}_{i=1}^m$ ($c =$ R, G and B), where $P_c^i \in \{0, \cdots, 255\}$ denotes the color component $c$ of the $i$-th pixel and $m = M \times N$. Then, the integer sequences are converted to three real-number sequences each of which corresponds to a different color component: $\{x_c^i(0)\}_{i=1}^m$, where

$$x_c^i(0) = x_{\min} + (x_{\max} - x_{\min}) \cdot P_c^i / 255. \tag{2}$$

[*] This paper has been published in *Chaos*, vol. 18, no. 3, article number 033112, 2008, DOI: 10.1063/1.2959102.

[†] Electronic address: `david.arroyo@iec.csic.es`

[‡] Current address: Fachbereich Informatik und Informationswissenschaft Universität Konstanz Fach M697, Universitätsstraße 10, 78457 Konstanz, Germany

After the above initialization process, the following encryption procedure is carried out separately for each color component to obtain the ciphertext:

1. Set $r = 1$.
2. Set the initial condition of the logistic map as follows:

$$x_0 = \begin{cases} x_c^m(r-1), & \text{if } i = 1, \\ x_c^{i-1}(r), & \text{if } 2 \leq i \leq m. \end{cases}$$

3. Iterate the chaotic logistic map from $x_0$ for $n$ times to obtain $x_n$.
4. Set $x_c^i(r) = x_n + x_c^i(r-1)$. If $x_c^i(r) > x_{\max}$, then subtract $(x_{\max} - x_{\min})$ from $x_c^i(r)$ to ensure $x_c^i(r) \in [x_{\min}, x_{\max}]$.
5. Set $r = r + 1$. If $r < j$, go to Step 2; otherwise the encryption procedure stops for the current color component.

After performing the above encryption procedure for all three color components, the three sequences $\{x_R^i(j)\}_{i=1}^m$, $\{x_G^i(j)\}_{i=1}^m$ and $\{x_B^i(j)\}_{i=1}^m$ make up the ciphertext.

As claimed in [9], the secret key is composed of the following four sub-keys:

1. The control parameter of the logistic map, i.e., $a$.
2. The image height and the image width, i.e., $M$ and $N$ respectively.
3. The number of chaotic iterations in Step 3, i.e., $n$.
4. The number of cycles, i.e., $j$.

The decryption procedure is similar to the above description, but in the reverse order, and the following inverse map

$$P_c^i = \text{round}[(x_c^i(0) - x_{\min}) \cdot 255/(x_{\max} - x_{\min})] \tag{3}$$

is used in the last step to recover the plain-image by converting real numbers back to integer pixel values. For more details about the encryption/decryption procedures, the reader is referred to [9].

## III. DESIGN PROBLEMS

### A. Key definition problems

Following Kerckhoffs' principle [10], the security of a cryptosystem should depend only on its key. For the cryptosystem defined in [9], the size of the image to be encrypted determines one of its four secret sub-keys. In a known-plaintext attack we have access to both the plain image and its encrypted version, which means that we know the size of the image. Moreover, in a ciphertext-only attack the value $m = M \times N$ is known and it is possible to get $M$ if $N$ is known and vice versa. Therefore, it is not a good idea to include the size of the image as part of the key, since it does not increase the difficulty to break the cryptosystem.

In addition, the control parameter $a$ of the logistic map is also part of the key. In [9] $a$ is chosen in $(3.57, 4)$ for the sake of the map defined in Eq. (1) being always chaotic. However, the bifurcation diagram of the logistic map (Fig. 1) shows the existence of periodic windows in that interval. It means that a user could choose $a$ such that the logistic map would be working in a non-chaotic area, which is not a good security criterium when considering chaotic cryptosystems [11, Rule 5]. Hence, it is advisable to give a more detailed definition of the possible values of $a$, so that the user can only choose those values of the control parameter $a$ preventing the logistic map from showing a periodic behavior.

Finally, the other parts of the key are the number of iterations of the logistic map per pixel ($n$) and the number of encryption cycles ($j$). As secret sub-keys, both values should possess a high level of entropy to avoid being guessed by a possible attacker. However, it is not advisable to select large values for $j$ and $n$, since it will definitely lead to a very slow encryption speed. On the other hand, using small values of $j$ and $n$ reduces the level of security, since those small values do not provide good confusion and diffusion properties. Both restrictions imply a reduction of the associated sub-key space and thus they make the brute-force attack more likely to be successful. As a conclusion, it is convenient to use $j$ and $n$ as design parameters and not as part of the secret key. This approach has been traditionally followed with respect to the number of encryption rounds in classical schemes such as DES and AES.

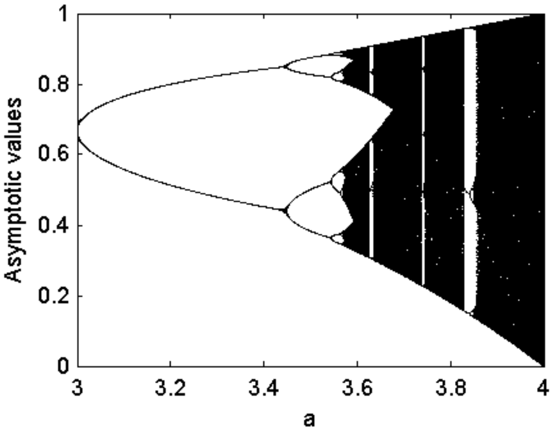


FIG. 1: Bifurcation diagram of the logistic map showing periodic windows.

### B. Underlying decryption error

As it happens during the encryption procedure, all the intermediate values $x_c^i(r)$ obtained through the decryption stage must be inside the phase space. This means that $x_{\min}$ should appear in Eq. (10) and (11) in [9] instead of 0. Having in mind this consideration, the performance of the decryption process will be analyzed in the following.

The cryptosystem described in [9] generates a ciphertext consisting of a number of real values. All the operations to encrypt an image are performed using floating-point arithmetic. From Section II we know that $x_c^i(r) = x_n + x_c^i(r-1)$, where $x_n$ is the resulting value of iterating the logistic map $n$ times from $x_0$. Hence, if we want to recover $x_c^i(r-1)$ (the original value of the $i$-th element in the last round), we have to iterate $n$ times the logistic map from $x_0$ to get $x_n$ and, after that, to substract this value from $x_c^i(r)$. However, the resulting value of this previous operation might not match the actual value of $x_c^i(r-1)$, due to the wobbling precision problem that exists when dealing with floating-point operations [12, p. 39]. This wobbling precision problem also causes the resulting guessed value of $x_c^i(r-1)$ to depend on the cryptosystem implementation. Therefore, if an image is encrypted on one platform and decrypted on another, and the implementations of floating-point arithmetics on both platforms are not compatible with each other, then the decrypted image might not match the original one. In [9] the cryptosystem was implemented using Microsoft Visual C# .NET 2005 and no comment was given about the wobbling precision problem in the decryption process. However, we have experimentally verified that this problem indeed exists when the cryptosystem is implemented using MATLAB on a PC with a 3 GHz processor and 2 GB of RAM. A very useful measure of the performance of the decryption procedure is the Mean Square Error or MSE. For $P$ and $P'$ being a plain image and the decrypted image respectively, the MSE for the color component $c$ is defined as

$$MSE_c = \sum_{i=1}^{m} (P_c^i - {P'}_c^i)^2/m, \tag{4}$$

where $c \in \{R, G, B\}$, $m = M \times N$ is the number of pixels of the images considered and the sequences $\{P_c^i\}_{i=1}^m$ and $\{P_c'\}_{i=1}^m$ are the result of scanning $P$ and $P'$ in the raster order. Consequently, for a well designed encryption/decryption scheme the MSE should be 0 for each color component. Unfortunately, for the cryptosystem under study, the values of MSE for all three color components are generally not equal to 0 due to the wobbling precision problem associated to the floating-point arithmetic.

In order to evaluate the underlying decryption error of the cryptosystem defined in [9], a 512 × 512 plain-image "Lena", as shown in Fig. 2, was encrypted and decrypted using the same key $(n, j, a) = (30, 1, 3.9)$. The results showed that the three MSEs obtained for the red, green and blue components of the decrypted image with respect to the original one were 6.49, 0.018, 0.057, respectively. For another key $(n, j, a) = (30, 3, 3.9)$, the obtained MSEs were 206.96, 123.45, 58.65, respectively. Figure 3 shows the decrypted image and the error image when the cryptosystem was implemented in MATLAB using a third key $(n, j, a) = (30, 5, 3.9)$. For this third situation, the resulting MSEs were 7439.49, 6324.15 and 4869.53.

FIG. 2: The plain-image “Lena”.

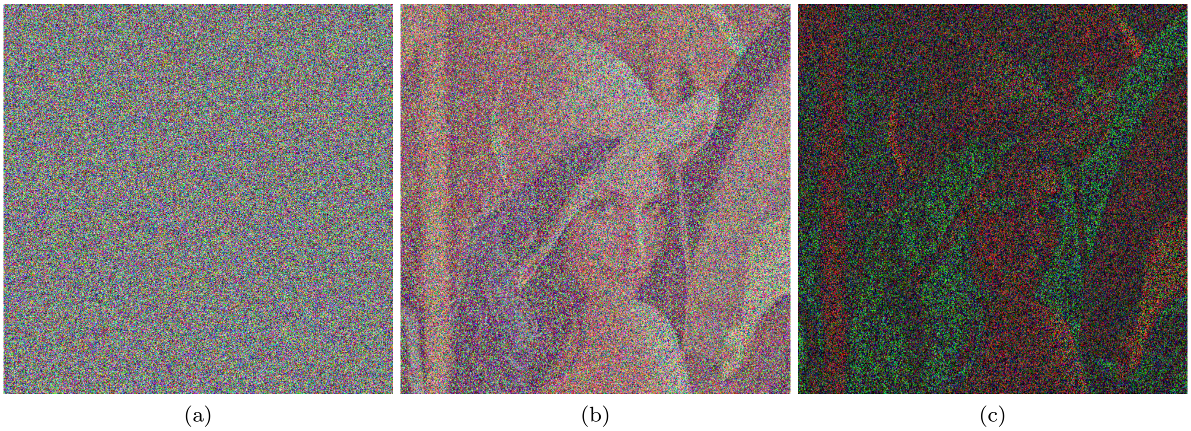


FIG. 3: Simulations with MATLAB (a) Ciphertext of the plain-image “Lena” (visualized as a pseudo-image by using Eq. (3)) (b) Recovered image of “Lena” using the same key (c) The error image between the original and the recovered “Lena”.

## IV. ATTACKS

### A. Control parameter estimation

The maximum value of $x_{n+1}$ in Eq. (1) is reached when $x_n = 0.5$, which informs that the maximum value of a sequence generated from the iteration of the logistic map is $a/4$, i.e., $x_{\max} = \max\left(\{x_i\}\right) \leq a/4$. The ciphertext of the cryptosystem proposed in [9] is composed of $3m$ real values, each of which is in the range $[x_{\min}, x_{\max}]$. This means that it is possible to approximate $x_{\max}$ as the maximum value of all the real values in the ciphertext, i.e.,

$$\hat{x}_{\max} = \max_{\substack{1\leq i\leq m\\ c=\text{R, G, B}}} x_c^i(j). \tag{5}$$

Then, from $x_{\max} \approx a/4$, one can estimate the secret value of the control parameter $a$ as

$$a \approx \hat{a} = 4 \cdot \hat{x}_{\max}. \tag{6}$$

Consequently, if we have a ciphertext, we can estimate the value of the sub-key $a$. In other words, a ciphertext-only attack allows us to estimate the sub-key $a$. In this sense, the image “Lena” (Fig. 2) was encrypted for $n = 20$, $j = 1$ and different values of $a \in [3.8, 4]$. These values of $a$ were then estimated from the ciphertexts by applying Eqs. (5) and (6). The parameter estimation error (PEE) was calculated as

$$PEE = |a - \hat{a}|, \tag{7}$$

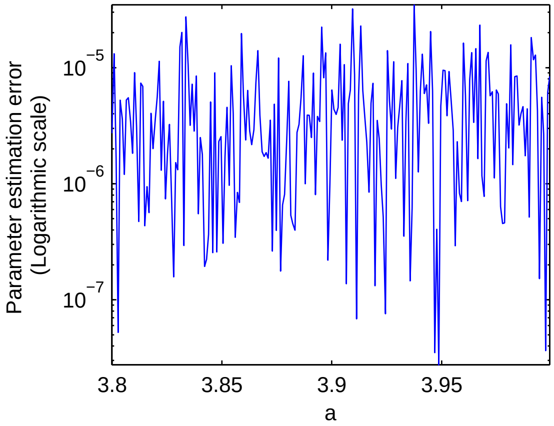


FIG. 4: Parameter estimation errors corresponding to the image “Lena”, when $n = 20$ and $j = 1$.

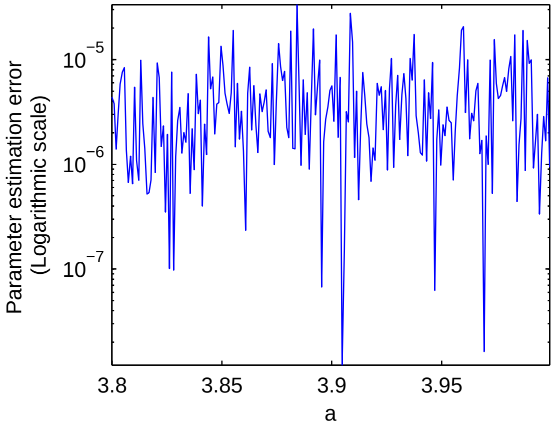


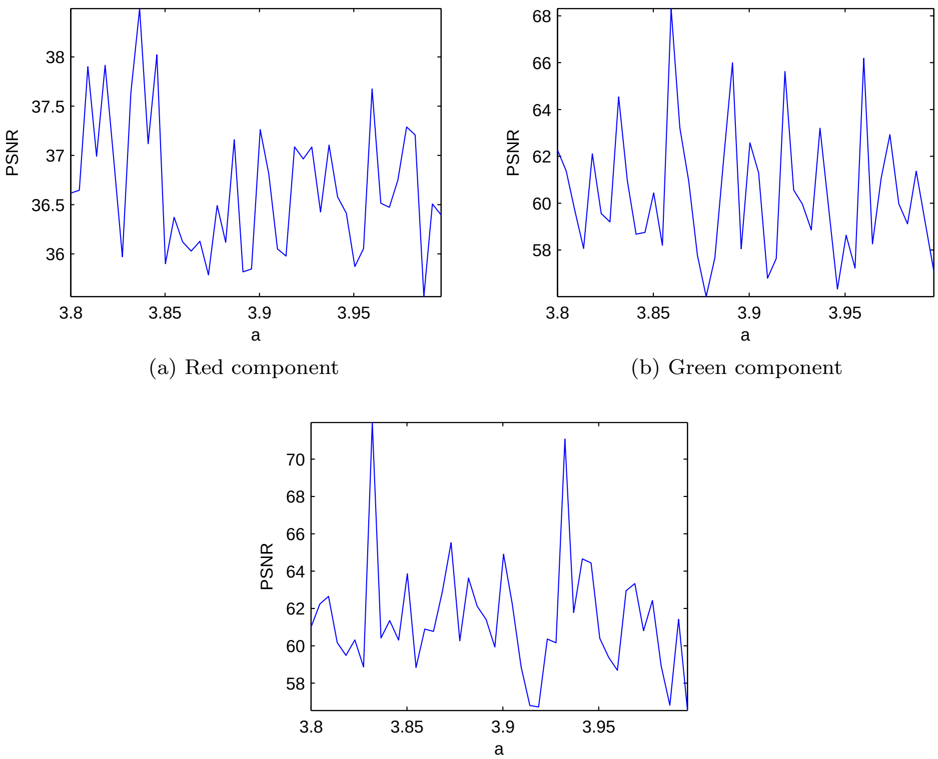
FIG. 5: Parameter estimation errors corresponding to the image “Lena”, when $n = 20$ and $j = 3$.

for different values of $a$ that were considered. The PEEs are shown in Fig. 4. The average estimation error was $5.236228 \times 10^{-6}$, whereas the maximum and minimum errors were $3.481322 \times 10^{-5}$ and $2.758853 \times 10^{-8}$, respectively. By increasing the value of $j$ from 1 to 3 and keeping the other sub-keys unchanged, the PEEs are shown in Fig. 5, being the mean estimation error $4.721420 \times 10^{-6}$, the minimum error $1.212016 \times 10^{-8}$ and the maximum error $3.355227 \times 10^{-5}$.

Finally, in Figs. 6 and 7 the sensitivity of the cryptosystem with respect to the control parameter $a$ is shown. This sensitivity is measured using the Peak Signal to Noise Ratio (PSNR), which is defined for the color component $c$ as

$$PSNR_c = 10 \cdot \log_{10} \left( \frac{255^2}{MSE_c} \right). \tag{8}$$

Figure 6 displays the PSNRs of the different color components of the decrypted image “Lena” with respect to the original image “Lena” for $a \in [3.8, 4]$ when the same key is used for encryption and decryption. The values of the other sub-keys are $n = 20$, $j = 3$. On the other hand, Figure 7 shows the PSNRs when the control parameter used in decryption shows some deviation from that employed in the encryption process. One can see that for a deviation of the control parameter of less than $10^{-10}$ and for a certain range of values of the control parameter, it is possible to recover the original image “Lena” with a similar PSNR to that obtained using the correct control parameter. For instance, for $a = 3.845621$ the PSNRs for the red, green and blue components of the recovered “Lena” are 35.899819, 60.437331 and 63.853450, respectively. For the same value of $a$ and a parameter estimation error equal to $10^{-12}$, the PSNR of the recovered “Lena” with respect to the original one is 17.480625 for the red component, 18.622578 for the green and 20.019512 for the blue component.



(a) Red component

(b) Green component

(c) Blue component

FIG. 6: PSNRs of the decrypted image "Lena" with respect to different values of the control parameter $a$.

## B. Timing attack

One important feature of a secure encryption scheme is that the encryption speed should not depend on the key value. Indeed, if the time consumed on encryption/decryption is correlated with the value of the key (or a sub-key), then it is possible to approximate that (sub-)key. This kind of attack is called timing attack [13, 14]. As it has been shown in Section II, in every encryption round, Step 3 is carried out through the $n$ iterations of Eq. (1), where $n$ is a sub-key. This means that, for a certain number of encryption rounds (i.e., a certain value of $j$) and a certain value of the control parameter $a$, the encryption speed decreases as $n$ does. Similarly, because the encryption/decryption procedure is composed of $j$ repeated cycles, the encryption speed will also become slower if the value of $j$ increases. To be more precise, for a given plain-image, we can expect the existence of the following bi-linear relationship between the encryption/decryption time (EDT) and the values of $n$ and $j$:

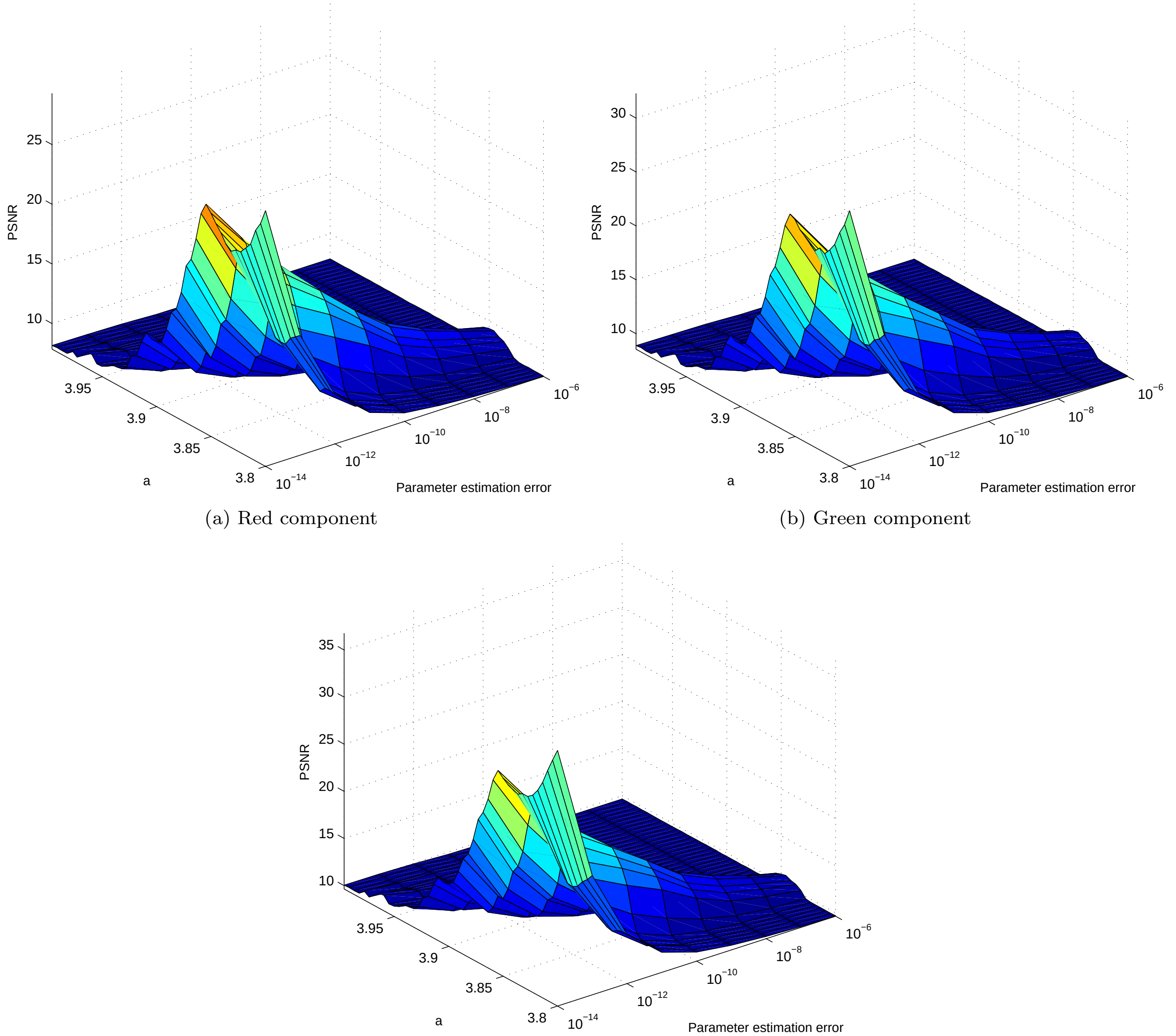
$$EDT(n, j) \approx (c \times n + d_0) \times j + d_1, \tag{9}$$

where $c$ corresponds to the common operations consumed on each chaotic iteration, $d_0$ to the operations performed in each cycle excluding those about chaotic iterations, and $d_1$ to those operations performed on the initialization process and the postprocessing after all the $j$ cycles are completed. In addition, because $a$ is just the control parameter of the chaotic map, it is expected that $EDT$ will be independent of its value.

With the aim of verifying this hypothesis, some experiments have been made under the following scenario. An image with random pixel values of size 256 × 256 was encrypted for different values of $a$, $n$ and $j$. The encryption time corresponding to each key is shown in Fig. 8, from which one can see that Eq. (9) is verified.

The above experimental results ensure the feasibility of a timing attack to a sub-key of the cryptosystem under study: by observing the encryption time, it is possible to estimate the values of $n$ if $j$ is known and vice versa. Without loss of generality, assuming an attacker Eve knows the value of $n$, but not that of $j$, let us demonstrate how the timing attack can be performed in practice. In this case, the relationship between EDT and the value of $j$ can be simplified as $EDT(n, j) = c_n \times j + d_n$, where $c_n = c \times n$ and $d_n = d_0 \times j + d_1$. Then, if Eve gets a temporary access to the encryption (or decryption) machine, she can carry out a real timing attack in the following steps:

1. She observes the whole process of encryption (or decryption) to get the encryption (or decryption) time $t_j$ and



(a) Red component

(b) Green component

(c) Blue component

FIG. 7: PSNRs of the decrypted image "Lena" with respect to different values of the control parameter $a$ and different parameter estimation errors.

also the size of the ciphertext (i.e., the size of the plaintext).

2. By choosing two keys with different values of $j$, she encrypts a plaintext (or decrypts a ciphertext) of the same size and gets $t_1$ and $t_2$.

3. She derives the values of $c_n$ and $d_n$ by substituting $t_1$ and $t_2$ into $EDT(n, j) = c_n \times j + d_n$.

4. She estimates the value of $j$ to be $\hat{j} = \text{round}((t_j - d_n)/c_n)$.

5. She verifies the estimated value $\hat{j}$ by using it to decrypt the observed ciphertext. If the recovered plaintext is something meaningful, the attack stops; otherwise, she turns to search the correct value of $j$ in a small neighborhood of $\hat{j}$ until a meaningful plaintext is obtained.

The above timing attack actually reveals that partial knowledge about the key constitutes useful information to determine the rest of the key. However, such a problem should not exist for a well-designed cryptosystem [11, Rule 7]. Hence, we reach the conclusion that the cryptosystem proposed in [9] was not well designed.

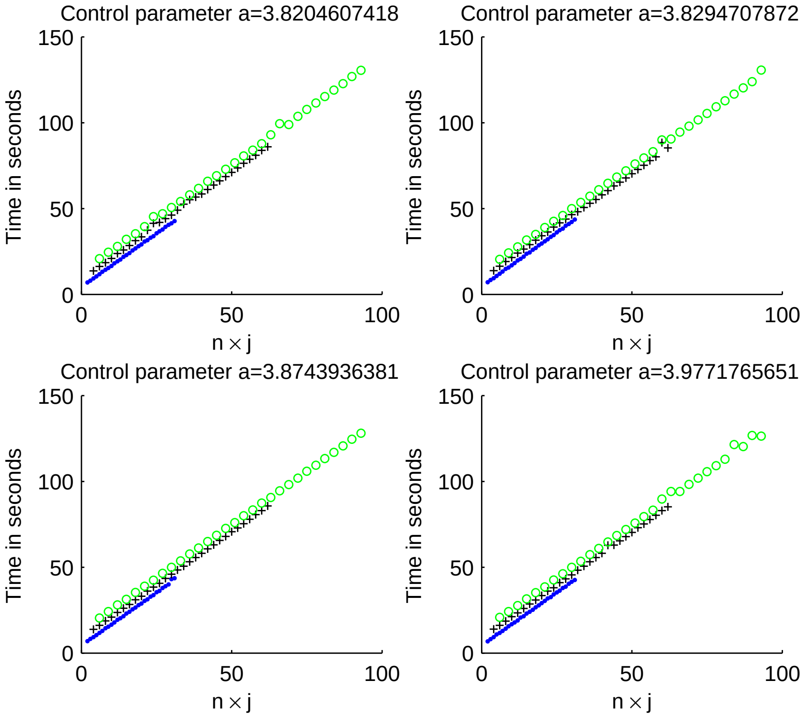


FIG. 8: Encryption time for images of size 256 × 256 and different values of the number of iterations $n$ and the number of encryption rounds.

Finally, it deserves being mentioned that the linear relationship between the encryption/decryption time and the value of $j$ has been implicitly shown in [9, Table I]. There, for an image of size 300 × 200 and $j$ equal to 1, 2 and 3, the encryption times were observed to be 13.6, 26.7 and 39.1 seconds, respectively. This clearly showed a linear relationship between the encryption time and the value of $j$. Unfortunately, the authors of [9] did not realize that this is a security defect that could be used to develop the timing attack reported in this paper.

## V. ENHANCEMENTS

To overcome the problems of the original cryptosystem, we propose to enhance it by applying the following rules:

- Use a piecewise linear chaotic map (PWLCM) [15] instead of the logistic map for the size of the chaotic phase space being independent with respect to the control parameter value. Indeed, the chaotic phase space of the PWLCM is (0,1) for all the values of the control parameter. The PWLCM also has a uniform invariant probability distribution function, which makes impossible to estimate the control parameter through the maximum value of the ciphertext, as we can do for the cryptosystem under study.
- The wobbling precision problem should be circumvented by forcing fixed-point computations. A possible solution is to transform the values of the phase space of the chaotic map into integer values, so the encryption and decryption operations are carried out using integer numbers instead of real numbers.
- Without loss of security, the enhanced cryptosystem should be easy to implement with acceptable cost and speed [11, Rule 3]. It is expected that the enhanced cryptosystem can encrypt at least a pixel per iteration to reach high encryption/decryption speed.
- The key of the enhanced cryptosystem should be precisely defined [11, Rule 4], and the key space from which valid keys are chosen should be precisely specified and avoid non-chaotic regions [11, Rule 5]. This can be assured by choosing the control parameter(s) of a PWLCM as the secret key, because for every valid control parameter, the behavior of the PWLCM is chaotic.

- Having in mind today's computer speed, the key space size should be $\kappa > 2^{100} = 10^{30}$ in order to elude brute-force attacks [11, Rule 15]. In the encryption scheme defined in [9] every color component is encrypted independently from the other color components. Nevertheless, the secret key employed in the encryption process of each color component is the same. It is convenient to use a different value of the key for each color component and make the encryption of the three color components dependent on each other, since this implies a considerable increase of the key space. It has been tested that the sensitivity of the PWLCM with respect to the control parameter is around $10^{-10}$. Therefore, when the control parameter is used as the key of the cryptosystem, the size of the key space will be $\kappa = 10^{10}$. Nonetheless, if we use a different parameter value for every color component, and the encryption of each color component depends on the others, the size of the key space will be $\kappa = 10^{30}$, which satisfies the security requirement related to the resistance against brute-force attacks.

## VI. CONCLUSIONS

In this paper, some problems of a new image encryption scheme based on chaotic map lattices are reported and two attacks on this cryptosystem have been presented. To overcome these problems and weaknesses, we have introduced some countermeasures to enhance the cryptosystem by following the cryptographical rules listed in [11].

## VII. ACKNOWLEDGMENTS

The work described in this paper was partially supported by *Ministerio de Educación y Ciencia of Spain*, research grant SEG2004-02418 and *Ministerio de Ciencia y Tecnologìa* of Spain, research grant TSI2007-62657. Shujun Li was supported by a research fellowship from the *Alexander von Humboldt Foundation, Germany.*

[1] M. S. Baptista, "Cryptography with chaos," Phys. Lett. A **240**, 50–54 (1998).

[2] S. Li, G. Chen, K.-W. Wong, X. Mou and Y. Cai, "Baptista-type chaotic cryptosystems: problems and countermeasures," Phys. Lett. A **332**, 368–375 (2004).

[3] E. Alvarez, A. Fernández, P. García, J. Jiménez and A. Marcano, "New approach to chaotic encryption," Phys. Lett. A **263**, 373–375 (1999).

[4] G. Chen, Y. Mao and C. K. Chui, "A symmetric image encryption based on 3D chaotic maps," Chaos Soliton Fractals **21**, 749–761 (2004).

[5] Z. H. Guan, F. Huang and W. Guan, "Chaos-based image encryption algorithm," Phys. Lett. A **346**, 153–157 (2005).

[6] N. Pareek, V. Patidar and K. K. Sud, "Image encryption using chaotic logistic map," Image Vis. Comput. **24**, 926–934 (2006).

[7] H. S. Kwok and W. K. S. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," Chaos Soliton Fractals **32**, 1518–1529 (2007).

[8] R. Rhouma, S. Meherzi and S. Belghith, "OCML-based colour image encryption," accepted by Chaos Solitons Fractals, in press, doi: 10.1016/j.chaos.2007.07.083 (2007).

[9] A. Pisarchik, N. Flores-Carmona and M.Carpio-Valadez, "Encryption and decryption of images with chaotic map lattices," Chaos **16**, art. no. 033118 (2006).

[10] A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography* (CRC Press, 1997).

[11] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," Int. J. Bifurc. Chaos **16**, 2129–2151 (2006).

[12] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. (SIAM, 1961).

[13] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," Advances in Cryptology – CRYPTO'96, vol. 1109 of *Lecture Notes in Computer Science*, 104–113 (1996).

[14] D. Brumley and D. Boneh, "Remote timing attacks are practical," Proceedings of the 12th USENIX Security Symposium, 1–14 (2003).

[15] S. Li, G. Chen and X. Mou, "On the dynamical degradation of digital piecewise linear chaotic maps," Int. J. Bifurc. Chaos **15**, 3119–3151 (2005).