Physics Letters A 372 (2008) 5973–5978

Contents lists available at ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

Cryptanalysis of a new image encryption algorithm based on hyper-chaos

Rhouma Rhouma *, Safya Belghith

6'com laboratory, Ecole Nationale d'Ingénieurs de Tunis (ENIT), Tunisia

ARTICLE INFO

Article history:
Received 23 February 2008
Received in revised form 12 June 2008
Accepted 23 July 2008
Available online 5 August 2008
Communicated by A.P. Fordy

Keywords:
Cryptanalysis
Chaotic encryption
Keystream
Hyper-chaos
Shuffle

ABSTRACT

This Letter proposes two different attacks on a recently proposed image based on hyper-chaos. The cryptosystem under study proceed first by shuffling the image rows and columns to disturb the high correlation among pixels by iterating the logistic map. Second, a keystream is generated to mix it with the pixels of the shuffled image using hyper-chaos. These two processes in the encryption stage present weakness, and a chosen plaintext attack and a chosen ciphertext attack can be done to recover the ciphered-image without any knowledge of the key value. It just demands three couples of plaintext/ciphertext to break totally the cryptosystem.

1. Introduction

Some researchers have pointed out that there exists tight relationship between chaos and cryptography [1–7]. Many fundamental characteristics of chaos, such as the ergodicity, mixing and exactness property and the sensitivity to initial conditions, can be connected with the "confusion" and "diffusion" property in cryptography. So it is a natural idea to use chaos to enrich the design of new ciphers. As a consequence, there have been proposed many chaotic ciphers in a very huge variety of design. We are interested on those dedicated to the image encryption. Image encryption is somehow different from text encryption due to some inherent features of image, such as bulk data capacity and high correlation among pixels. So far, many chaos-based image cryptosystems have been proposed [8–18]. Although a number of them have been cryptanalyzed, many others have not been effectively attacked like the one in [8]. In this Letter, we propose to break the image encryption algorithm proposed by T. Gao and Z. Chen in [8]. First, this paper gives a detailed introduction of the cryptosystem, as a basis of the whole Letter. The image encryption scheme under study consists of two parts: The image encryption based on total shuffling matrix, and the mixing operation of the shuffled image with a keystream generated from a hyper-chaotic system. First, an image of size $N \times M$ is considered, every pixel of this plain image is noted $P_{i,j}$, with $i = 0, \ldots, M-1$ and $j = 0, \ldots, N-1$. By using the logistic map given by Eq. (1) departing from an initial condition x_0:

$$x_{n+1} = 4x_n(1 - x_n). \tag{1}$$

After some iterations n, a new x_0 is derived from the final iteration x_n and a number h_i is calculated:

$$h_i = \operatorname{mod}(x_0 \times 10^{14}, M). \tag{2}$$

The iteration of the logistic map will continue until getting M different data between 0 and $M-1$ noted h_i, $i = 0, \ldots, M-1$. Then rearrange the rows of the plain image P according to $\{h_i,\ i = 0, \ldots, M-1\}$. h_i will be the ith row in the shuffled rows image noted P^h. Then, this process is repeated to shuffle the column position of every row in P^h to obtain a totally shuffled image in rows and columns P^{hl}. The equation used to calculate the position of the shuffled column of every row is:

$$l_{i,j} = \operatorname{mod}(x_0 \times 10^{14}, N) \tag{3}$$

for every column $i = 1, \ldots, M$ and row $j = 1, \ldots, N$.

Second, an hyper-chaotic system given by (4) is used:

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1), \\ \dot{x}_2 = -x_1x_3 + dx_1 + cx_2 - x_4, \\ \dot{x}_3 = x_1x_2 - bx_3, \\ \dot{x}_4 = x_1 + k. \end{cases} \tag{4}$$

The encryption scheme is based on the combination of state variables of the above hyper-chaotic system according to these steps:

(1) First, the system on (4) is iterated for N_0 times.

* Corresponding author.
E-mail addresses: rhoouma@yahoo.fr (R. Rhouma), safya.belghith@enit.rnu.tn (S. Belghith).

0375-9601/$ – see front matter
doi:10.1016/j.physleta.2008.07.057

(2) Four variables are generated from the hyper-chaotic system and then transformed to integers applying the following:

$$x_i = \mathrm{mod}\big(\big(|x_0| - \mathrm{Floor}\big(|x_0|\big)\big) \times 10^{14}, 256\big). \tag{5}$$

(3) Generate $\bar{x}_1$ using the following:

$$\bar{x}_1 = \mathrm{mod}(x_1, 4). \tag{6}$$

According to the value of $\bar{x}_1$, three variables (B_1, B_2, B_3) from the four variables (x_1, x_2, x_3, x_4) generated from (5) are chosen to perform encryption operation using an association table (for more details to perform this step, the reader is advised to see Table 2 in Ref. [8]). And then, three pixels from the plain shuffled image P^{hl} are mixed with the keystream B_k, $k = 1, \ldots, 3$, like the following:

$$\begin{cases} C_{3\times(i-1)+1} = P_{3\times(i-1)+1} \oplus B_1, \\ C_{3\times(i-1)+2} = P_{3\times(i-1)+2} \oplus B_2, \\ C_{3\times(i-1)+3} = P_{3\times(i-1)+3} \oplus B_3. \end{cases} \tag{7}$$

P_i and C_i, $i = 1, 2, \ldots, N \times M$, represent the pixel of the plain shuffled image P^{hl} and the ciphered image C, respectively.

(4) Continue on iterating the hyper-chaotic system, and go to step (2) until the whole image is totally ciphered.

The decryption algorithm is similar to the encryption algorithm. That is, for the encrypted image, firstly, decrypt the image using hyper-chaotic system with the same parameters and initial values as that used in encryption, and then anti-shuffle the resulting image, we will get the original image. As claimed in [8], the initial values of Logistic map and hyper-chaotic system are used as secret keys. For more details, the reader is referred to [8].

2. Classical types of attacks

When cryptanalyzing a cryptosystem, the general assumption made is that the cryptanalyst knows exactly the design and working of the cryptosystem under study, i.e., he knows everything about the cryptosystem except the secret key. This is an evident requirement in today's secure communications networks, usually referred to as Kerchoff's principle [19]. There are four classical types of attacks and it is possible to differentiate between different levels of these attacks based on the level of knowledge of the attacker to the cryptosystem and if or not he has the encryption/decryption machinery or knowledge of some couple of plaintext/ciphertext. So, we enumerate them ordered from the hardest types of attack to the easiest:

(1) Ciphertext only: the opponent possesses just a string of ciphertext.
(2) Known plaintext: the opponent possesses a string of plaintext, M, and the corresponding ciphertext, D.
(3) Chosen plaintext: the opponent has obtained temporary access to the encryption machinery. Hence he can choose a plaintext string, M, and construct the corresponding ciphertext string, D.
(4) Chosen ciphertext: the opponent has obtained temporary access to the decryption machinery. Hence he can choose a ciphertext string, D, and construct the corresponding plaintext string, M.

In each of these four attacks, the objective is to determine the key that was used. It suffices that one of the attacks is successful to consider an algorithm insecure.

3. Weakness of the cryptosystem based on the hyper-chaotic map

The cipher under study behaves as a stream cipher [19]. The operation of the algorithm as a stream cipher can be explained as follows. Assume that K is the key, given by initial conditions of the hyper-chaotic system and that P composed by P_i is the plaintext. A keystream $B = B_1 B_2 \ldots$ is generated using Eqs. (4) and (6). This keystream is used to encrypt the plaintext according to the rule:

$$C = e_{B_1}(P_1) e_{B_2}(P_2) \ldots = C_1 C_2 \ldots. \tag{8}$$

Decrypting the ciphertext string C can be accomplished by computing the keystream B given the knowledge of the key K and undoing the operations e_{B_i}.

The most serious problem of this cryptosystem is to make the generation of the keystream the same for every plaintext/ciphertext. Next, it is shown how to recover the keystream using chosen ciphertext and chosen plaintext attacks. We note that knowing the keystream B generated by a certain key K is entirely equivalent to knowing the key [20]. Moreover, the shuffling process (1st process of the encryption procedure) of the plain image is weak and can be guessed with a chosen plaintext and chosen ciphertext attacks.

3.1. Chosen plaintext attack CPA

Assume that we have a ciphertext $C = C_1 C_2 \ldots$ (the ciphered image written as a vector of length $N \times M$), to decrypt without knowing the key K. We assume that we have obtained temporary access to the encryption machinery. We describe the steps leading to recover the plain image P from the ciphered image C:

(1) We request the ciphertext of the plaintext $M = m_1 m_2 \ldots = 00000\ldots$: a plaintext of the same size of the ciphertext C constructed by the pixels of values $m_i = 0$ for every $i = 1, 2, \ldots, N \times M$. We obtain the ciphertext $D = d_1 d_2 \ldots$.
The keystream $B = B_1 B_2 \ldots$ can be generated from D by:

$$B_i = m_i \oplus d_i = d_i \tag{9}$$

for every $i = 1, 2, \ldots, N \times M$.
The recovered shuffled image $P^{hl} = P_1^{hl} P_2^{hl} \ldots$ can be obtained using the calculated keystream B and the ciphertext C:

$$P_i^{hl} = C_i \oplus B_i \tag{10}$$

for every $i = 1, 2, \ldots, N \times M$.

(2) We request now the ciphertext of an image $M \times N$ noted J whose all the rows of its first column is composed by the value pixel 1, all the rows of the second column is composed by 2, and so on, until the last column N whose all its rows is composed with the value N:

$$J = \begin{pmatrix} 1 & 2 & \ldots & N \\ 1 & 2 & \ldots & N \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \ldots & N \end{pmatrix}. \tag{11}$$

To show an example, we will consider that $M = N = 4$, so

$$J = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}.$$

The corresponding ciphered image is noted $J^c = J_1^c J_2^c \ldots$. With the calculated keystream B in step (1), we generate the shuffled image $J^{hl} = J_1 J_2 \ldots$ of J by applying the following:

$$J_i = J_i^c \oplus B_i. \tag{12}$$

With the given example, we find that

$$J^{hl} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \\ 2 & 1 & 3 & 4 \\ 4 & 2 & 3 & 1 \end{pmatrix}.$$

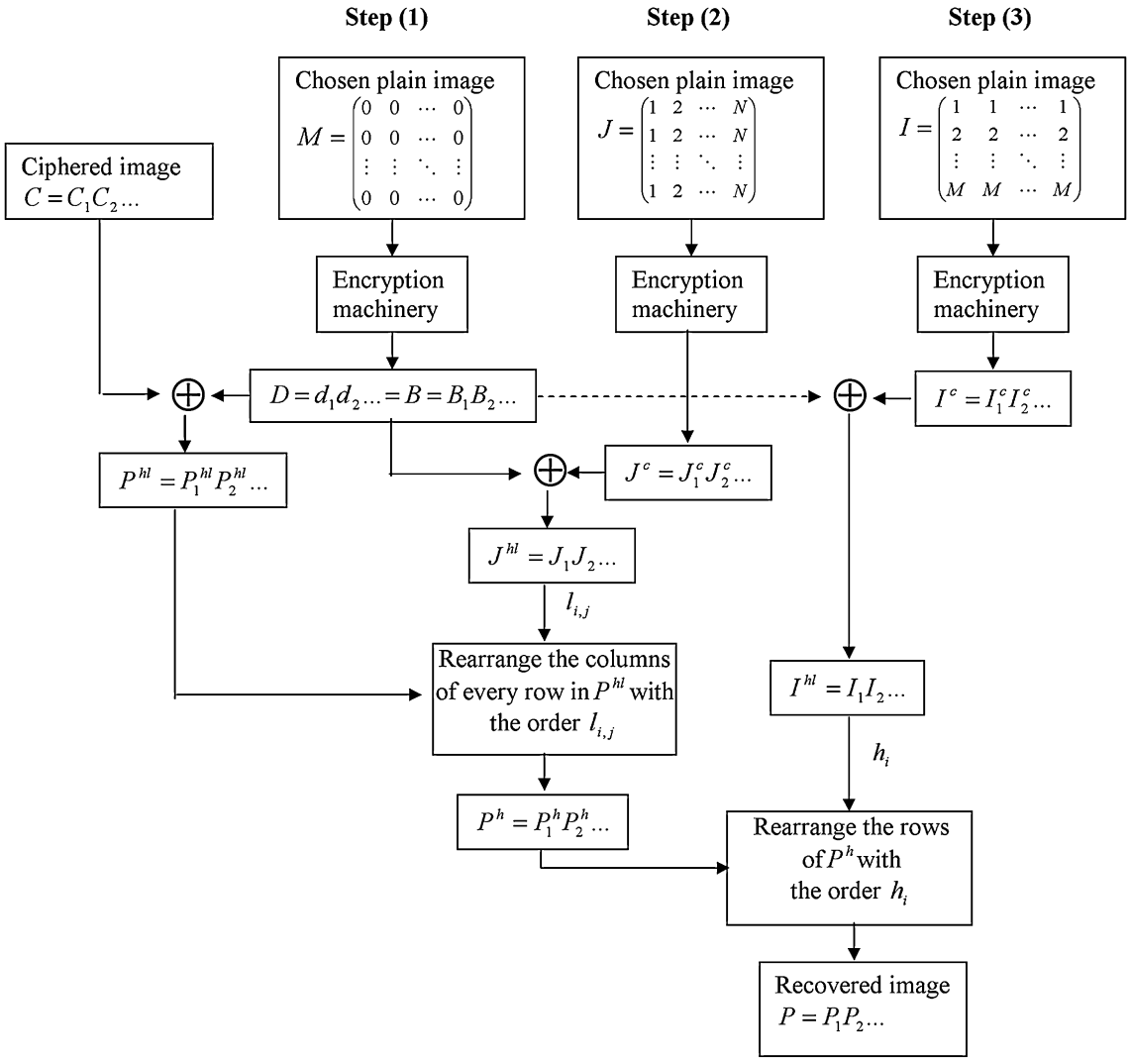

Fig. 1. Chosen plaintext attack.

So, for a 4×4 matrix, for the first row, the columns were ordered in the form of $\{l_{i,1},\ i = 1, \ldots, N\} = 1, 2, 3, 4$. For the second row, the columns were ordered in the form of $\{l_{i,2},\ i = 1, \ldots, N\} = 1, 4, 3, 2$. For the third row, the columns were ordered in the form of $\{l_{i,3},\ i = 1, \ldots, N\} = 2, 1, 3, 4$. And for the last row, the columns were ordered in the form of $\{l_{i,4},\ i = 1, \ldots, N\} = 4, 2, 3, 1$. We will use these $l_{i,j}$ to generate the image P^h by rearranging the columns of every row in the shuffled image P^{hl} which was recovered from the ciphered image in step (1). This can be generalized for any matrix with rows less or equal than 256.

(3) We request now the ciphertext of an image $M \times N$ noted I whose all the columns of its first row is composed by the value pixel 1, all the columns of the second row is composed by 2, and so on, until the last row M whose all its columns is composed with the value M:

$$I = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ 2 & 2 & \ldots & 2 \\ \vdots & \vdots & \ddots & \vdots \\ M & M & \ldots & M \end{pmatrix}. \tag{13}$$

To show an example, we will consider that $M = N = 4$, so

$$I = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}.$$

The corresponding ciphered image is noted $I^c = I_1^c I_2^c \ldots$. With the calculated keystream B in step (1), we generate the shuffled image $I^{hl} = I_1 I_2 \ldots$ of I by applying the following:

$$I_i = I_i^c \oplus B_i \tag{14}$$

With the given example, we find that

$$I^{hl} = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

One can verify easily that $I^h = I^{hl}$ because the columns values are the same for every row. So, for a matrix composed by 4 rows, the rows were ordered in the form of $\{h_i,\ i = 1, \ldots, M\} = 4, 3, 2, 1$. We will use these h_i to generate the recovered image P by rearranging the rows of the image P^h recovered from step (2).

Fig. 1 gives a graphic description of the chosen plaintext attack. Table 1 gives a detailed description of the CPA steps on a ciphered image of size 4×4. In Fig. 2, we show the simulations results of a Chosen plaintext attack on the Ciphered Lena image of size 256×256.

Table 1
Steps leading to recover the plain image P of size 4×4 from the ciphered image C in a CPA scenario

Plain image P	Ciphered image C	Step 1		
		Chosen plain image M	Keystream $B = D$ ciphered image of M	Recovered shuffled image P^{hl}
$\begin{pmatrix} 114 & 192 & 86 & 156 \\ 114 & 130 & 130 & 145 \\ 102 & 99 & 76 & 211 \\ 76 & 136 & 218 & 86 \end{pmatrix}$	$\begin{pmatrix} 165 & 226 & 4 & 106 \\ 115 & 203 & 247 & 156 \\ 154 & 56 & 149 & 92 \\ 44 & 24 & 9 & 196 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 233 & 106 & 222 & 60 \\ 21 & 24 & 187 & 255 \\ 24 & 74 & 23 & 205 \\ 176 & 216 & 95 & 182 \end{pmatrix}$	$\begin{pmatrix} 76 & 136 & 218 & 86 \\ 102 & 211 & 76 & 99 \\ 130 & 114 & 130 & 145 \\ 156 & 192 & 86 & 114 \end{pmatrix}$
Step 2				
Chosen plain image J	Ciphered image J^c	Shuffled image J^{hl}	$l_{i,j}$	Recovered image P^h
$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$	$\begin{pmatrix} 232 & 104 & 221 & 56 \\ 20 & 28 & 184 & 253 \\ 26 & 75 & 20 & 201 \\ 180 & 218 & 92 & 183 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \\ 2 & 1 & 3 & 4 \\ 4 & 2 & 3 & 1 \end{pmatrix}$	$\{l_{i,1}\} = 1, 2, 3, 4$ $\{l_{i,2}\} = 1, 4, 3, 2$ $\{l_{i,3}\} = 2, 1, 3, 4$ $\{l_{i,4}\} = 4, 2, 3, 1$	$\begin{pmatrix} 76 & 136 & 218 & 86 \\ 102 & 99 & 76 & 211 \\ 114 & 130 & 130 & 145 \\ 114 & 192 & 86 & 156 \end{pmatrix}$
Step 3				
Chosen plain image I	Ciphered image I^c	Shuffled image $I^{hl} = I^h$	h_i	Recovered image P
$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$	$\begin{pmatrix} 237 & 110 & 218 & 56 \\ 22 & 27 & 184 & 252 \\ 26 & 72 & 21 & 207 \\ 177 & 217 & 94 & 183 \end{pmatrix}$	$\begin{pmatrix} 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix}$	$\{h_i\} = 4, 3, 2, 1$	$\begin{pmatrix} 114 & 192 & 86 & 156 \\ 114 & 130 & 130 & 145 \\ 102 & 99 & 76 & 211 \\ 76 & 136 & 218 & 86 \end{pmatrix}$

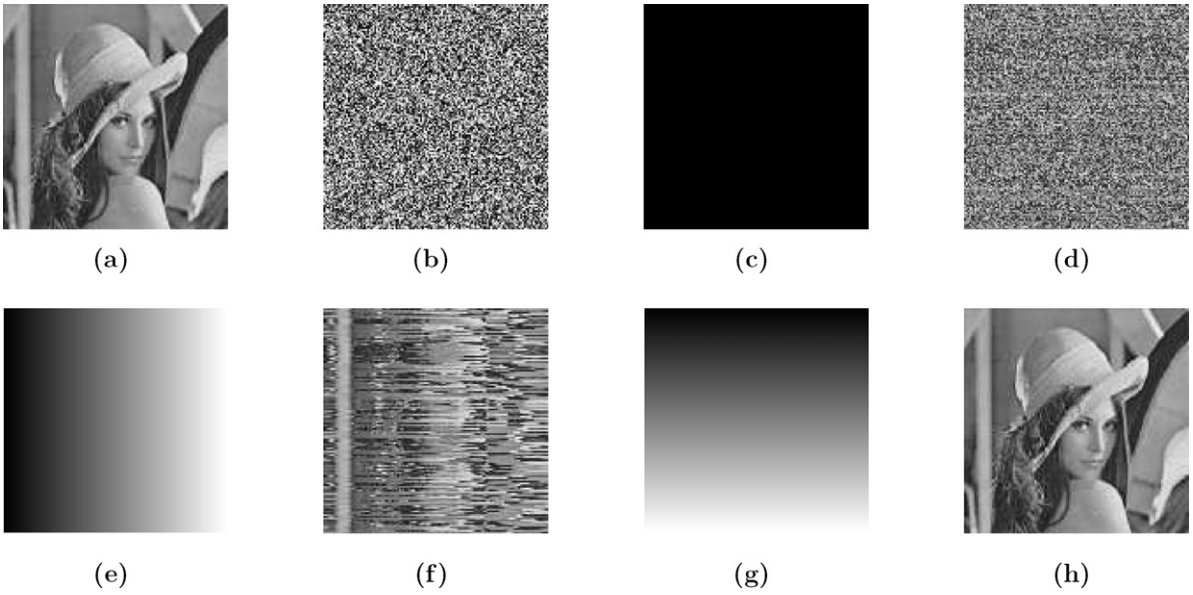

Fig. 2. (a) Original image of Lena P; (b) Ciphered image C; (c) Chosen plain image $M = \text{zeros}(M, N)$; (d) Shuffled image P^{hl}; (e) Chosen plain image J; (f) Shuffled image P^h; (g) Chosen plain image I; (h) Recovered image of Lena P.

3.2. Chosen ciphertext attack CCA

Assume that we have a ciphertext $C = C_1 C_2 \ldots$, to decrypt without knowing the key K. We assume that we have obtained temporary access to the decryption machinery. We describe the steps leading to recover the plain image P from the ciphered image C:

(1) We request the plaintext of the ciphertext $D = d_1 d_2 \ldots = 00000 \ldots$: a ciphertext of the same size of the ciphertext C constructed by the pixels of values $d_i = 0$ for every $i = 1, 2, \ldots, N \times M$. We obtain the plaintext $M = m_1 m_2 \ldots$.
The keystream $B = B_1 B_2 \ldots$ can be generated from M by:

$$B_i = m_i \oplus d_i = m_i \tag{15}$$

for every $i = 1, 2, \ldots, N \times M$.
The recovered shuffled image $P^{hl} = P_1^{hl} P_2^{hl} \ldots$ can be obtained using the calculated keystream B and the ciphertext C:

$$P_i^{hl} = C_i \oplus B_i \tag{16}$$

for every $i = 1, 2, \ldots, N \times M$.

(2) We construct an image I^{hl} of size $M \times N$ whose all the columns of its first row is composed by the value pixel 1, all the columns of the second row is composed by 2, and so on, until the last row M whose all its columns is composed with the value M. To show an example, we will consider that $M = N = 4$, so

$$I^{hl} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}.$$

We request now the plaintext of $I^c = I^{hl} \oplus B$, B was calculated in step (1). The corresponding plain image is noted $I = I_1 I_2 \ldots$. With the given example, we find that

$$I = \begin{pmatrix} 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \end{pmatrix}.$$

So, for a matrix composed by 4 rows, the rows were ordered in the form of $\{h_i,\ i = 1, \ldots, M\} = 3, 4, 1, 2$.

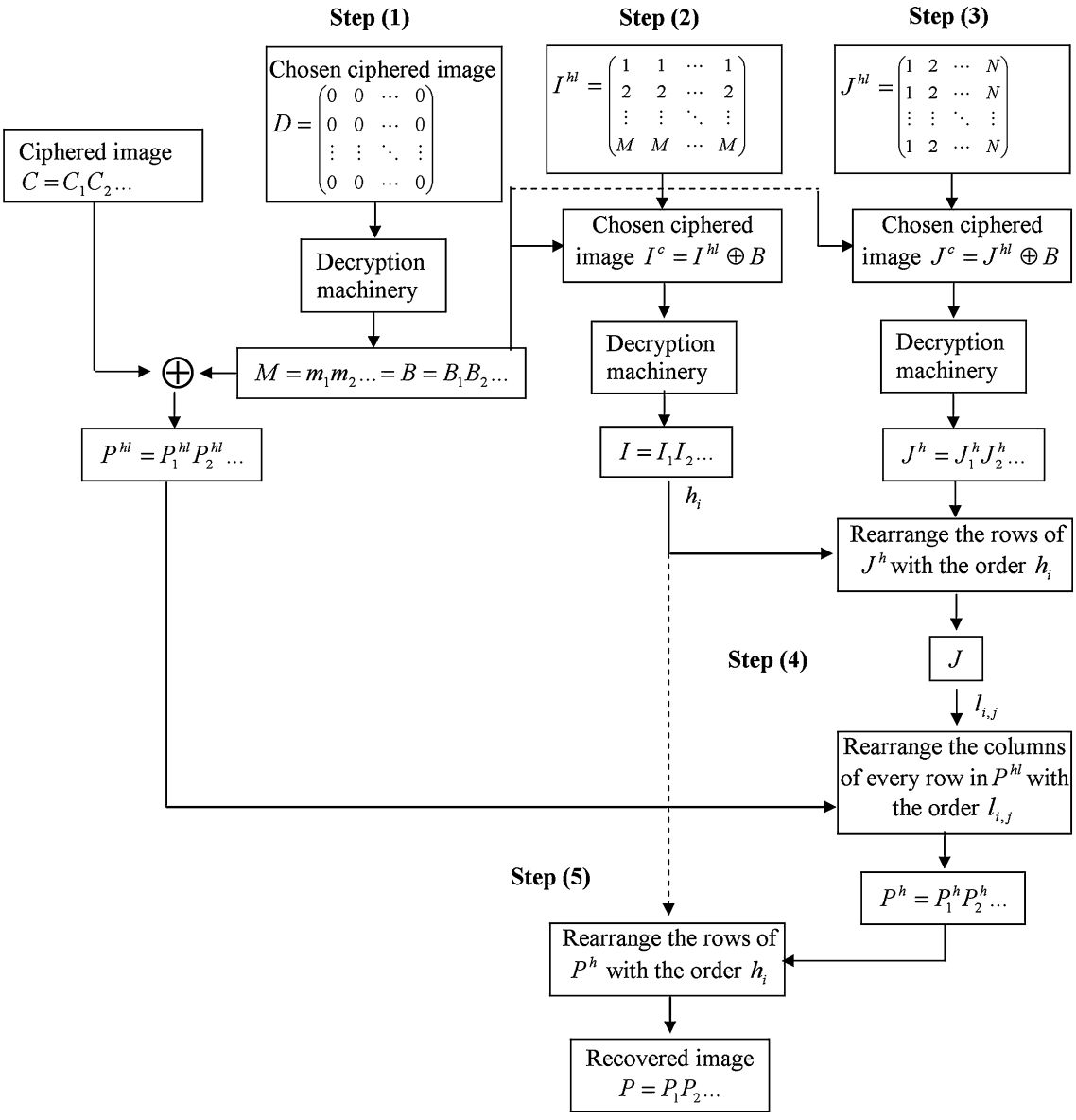

Fig. 3. Chosen ciphertext attack.

(3) We construct an image J^{hl} of size $M \times N$ whose all the rows of its first column is composed by the value pixel 1, all the rows of the second column is composed by 2, and so on, until the last column N whose all its rows is composed with the value N. To show an example, we will consider that $M = N = 4$, so

$$J^{hl} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}.$$

We request now the plaintext of the image $J^c = J^{hl} \oplus B$, B was calculated in step (1). The corresponding plain image is noted $J^h = J_1^h J_2^h \ldots$. With the given example, we find that

$$J^h = \begin{pmatrix} 3 & 2 & 4 & 1 \\ 1 & 3 & 4 & 2 \\ 2 & 1 & 4 & 3 \\ 2 & 4 & 1 & 3 \end{pmatrix}.$$

(4) And then, we will use the result of step (2) to rearrange the rows of J using the calculated $\{h_i,\ i = 1, \ldots, M\} = 3, 4, 1, 2$, we will find the matrix

$$J = \begin{pmatrix} 2 & 1 & 4 & 3 \\ 2 & 4 & 1 & 3 \\ 3 & 2 & 4 & 1 \\ 1 & 3 & 4 & 2 \end{pmatrix}.$$

So, for a 4×4 matrix, for the first row, the columns were ordered in the form of $\{l_{i,1},\ i = 1, \ldots, N\} = 2, 1, 4, 3$. For the second row, the columns were ordered in the form of $\{l_{i,2},\ i = 1, \ldots, N\} = 2, 4, 1, 3$. For the third row, the columns were ordered in the form of $\{l_{i,3},\ i = 1, \ldots, N\} = 3, 2, 4, 1$. And for the last row, the columns were ordered in the form of $\{l_{i,4},\ i = 1, \ldots, N\} = 1, 3, 4, 2$. These $l_{i,j}$ are then used to reconstruct the order of the columns of the shuffled image P^{hl} found in step (1). We find the image P^h.

(5) We will use the result of step (2) to rearrange the rows of P^h using the calculated $\{h_i,\ i = 1, \ldots, M\} = 3, 4, 1, 2$, we will find the recovered image P.

Table 2
Steps leading to recover the plain image P of size 4×4 from the ciphered image C in a CCA scenario

Plain image P	Ciphered image C	Step 1		
		Chosen ciphered image D	Keystream $B = M$ plain image of D	Recovered shuffled image P^{hl}
$\begin{pmatrix} 114 & 192 & 86 & 156 \\ 114 & 130 & 130 & 145 \\ 102 & 99 & 76 & 211 \\ 76 & 136 & 218 & 86 \end{pmatrix}$	$\begin{pmatrix} 165 & 226 & 4 & 106 \\ 115 & 203 & 247 & 156 \\ 154 & 56 & 149 & 92 \\ 44 & 24 & 9 & 196 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 233 & 106 & 222 & 60 \\ 21 & 24 & 187 & 255 \\ 24 & 74 & 23 & 205 \\ 176 & 216 & 95 & 182 \end{pmatrix}$	$\begin{pmatrix} 76 & 136 & 218 & 86 \\ 102 & 211 & 76 & 99 \\ 130 & 114 & 130 & 145 \\ 156 & 192 & 86 & 114 \end{pmatrix}$

Step 2			
Constructed image I^{hl}	Chosen ciphered image $I^c = I^{hl} \oplus B$	Plain image $I = I^h$	h_i
$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix}$	$\begin{pmatrix} 232 & 107 & 223 & 61 \\ 23 & 26 & 185 & 253 \\ 27 & 73 & 20 & 206 \\ 180 & 220 & 91 & 178 \end{pmatrix}$	$\begin{pmatrix} 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 \\ 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix}$	$\{h_i\} = 4, 3, 2, 1$

Step 3		
Constructed image J^{hl}	Chosen ciphered image $J^c = J^{hl} \oplus B$	Shuffled image J^h
$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}$	$\begin{pmatrix} 232 & 104 & 221 & 56 \\ 20 & 26 & 184 & 251 \\ 25 & 72 & 20 & 201 \\ 177 & 218 & 92 & 178 \end{pmatrix}$	$\begin{pmatrix} 4 & 2 & 3 & 1 \\ 2 & 1 & 3 & 4 \\ 1 & 4 & 3 & 2 \\ 1 & 2 & 3 & 4 \end{pmatrix}$

Step 4			Step 5
Plain image J	$l_{i,j}$	Recovered image P^h	Recovered plain image P
$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \\ 2 & 1 & 3 & 4 \\ 4 & 2 & 3 & 1 \end{pmatrix}$	$\{l_{i,1}\} = 1, 2, 3, 4$ $\{l_{i,2}\} = 1, 4, 3, 2$ $\{l_{i,3}\} = 2, 1, 3, 4$ $\{l_{i,4}\} = 4, 2, 3, 1$	$\begin{pmatrix} 76 & 136 & 218 & 86 \\ 102 & 99 & 76 & 211 \\ 114 & 130 & 130 & 145 \\ 114 & 192 & 86 & 156 \end{pmatrix}$	$\begin{pmatrix} 114 & 192 & 86 & 156 \\ 114 & 130 & 130 & 145 \\ 102 & 99 & 76 & 211 \\ 76 & 136 & 218 & 86 \end{pmatrix}$

Fig. 3 gives a graphic description of the chosen ciphertext attack. Table 2 gives a detailed description of the CCA steps on a ciphered image of size 4×4.

4. Conclusion

In this Letter, two attacks were presented to break a recently chaotic cryptosystem based on a hyper-chaotic system. It has been shown that the reuse of the keystream more than once makes it weak against chosen ciphertext, chosen plaintext attacks. The generated keystream neither depends on the plaintext nor does it depend on the ciphertext, which makes it unchangeable in every encryption process. Moreover, the process of shuffling the image is predictable and we can extract the original image from the shuffled image. Three couples of plaintext/ciphertext were enough to break the cryptosystem in a chosen ciphertext and chosen plaintext attacks scenarios. Note that in the case where the cryptosystem make the keystream changes in every encryption procedure, that will make it secure and the described attacks will be harmless to the algorithm. Changing the keystream either means (1) to change the key in every encryption procedure and by that the cryptosystem will be more like a one time pad [19] which is a kind of a stream cipher whose never reuse its key, in this case, the cryptosystem will be totally secure but we have to affront the distribution key problem and many others issues that make the one time pad impractical in a real secure communication. (2) or to make the generation of the keystream dependent to the plaintext or the ciphertext, and that will make the keystream change for every encryption procedure without changing the key. Doing so, the cryptosystem will be designed in a CBC (Cipher-Block Chaining [21]) or PCBC (Propagating Cipher-Block Chaining [22]) mode of block encryption. This solution is more adequate than to change the cryptosystem as a one time pad because it is secure enough and practical in the same time.

References

[1] R. Brown, L.O. Chua, Int. J. Bifur. Chaos 6 (2) (1996) 219.
[2] J. Fridrich, Int. J. Bifur. Chaos 8 (6) (1998) 1259.
[3] L. Kocarev, G. Jakimoski, T. Stojanovski, U. Parlitz, From chaotic maps to encryption schemes, in: Proceedings of the IEEE International Symposium Circuits and Systems 98, vol. 4, IEEE, 1998, pp. 514–517.
[4] G. Alvarez, F. Monotoya, G. Pastor, M. Romera, Chaotic cryptosystems, in: Proceedings of the International IEEE Carnahan Conference on Security Technology, IEEE, 1999, pp. 332–338.
[5] M. Gotz, K. Kelber, W. Schwarz, IEEE Trans. Circuits Syst. I 44 (10) (1997) 963.
[6] S. Li, X. Mou, Y. Cai, Pseudo-random bit generator based on couple chaotic systems and its application in stream-ciphers cryptography, in: Progress in Cryptology—INDOCRYPT 2001, in: Lecture Notes in Computer Science, vol. 2247, Springer-Verlag, Berlin, 2001, pp. 316–329.
[7] S. Li, Analyses and new designs of digital chaotic ciphers, PhD thesis, Jiaotong University China, 2003.
[8] T. Gao, Z. Chen, Phys. Lett. A 372 (2008) 394.
[9] Z.H. Guan, F. Huang, W. Guan, Phys. Lett. A 346 (2005) 153.
[10] S. Behnia, A. Akhshani, H. Mahmodi, A. Akhavan, Chaos Solitons Fractals 35 (2) (2008) 408.
[11] N.K. Pareek, V. Patidar, K.K. Sud, Ima. Vis. Comput. 24 (2006) 926.
[12] H.S. Kwok, W.K.S. Tang, Chaos Soliton Fractals 32 (2007) 1518.
[13] J.C. Yen, J.I. Guo, A new chaotic key based design for image encryption and decryption, in: Proceedings of the IEEE International Symposium Circuits and Systems, vol. 4, 2000, pp. 49–52.
[14] S. Li, X. Zheng, X. Mou, Y. Cai, Chaotic encryption scheme for real time digital video, in: Proceedings of SPIE on Electronic Imaging, San Jose, CA, USA, 2002.
[15] G. Chen, Y. Mao, C.K. Chui, Chaos Soliton Fractals 21 (2004) 749.
[16] S. Lian, Efficient image or video encryption based on spatiotemporal chaos system, Chaos Solitons Fractals (2007), doi:10.1016/j.chaos.2007.10.054.
[17] T. Xiang, K.-w. Wong, X. Liao, Chaos 17 (2007) 023115.
[18] R. Rhouma, S. Meherzi, S. Belghith, OCML-based colour image encryption, Chaos Soliton Fractals (2007), doi:10.1016/j.chaos.2007.07.083.
[19] D.R. Stinson, Cryptography: Theory and Practice, CRC Press, Boca Raton, FL, 1995.
[20] G. Alvarez, F. Montoya, M. Romera, G. Pastor, Phys. Lett. A 311 (2–3) (2003) 172.
[21] W.F. Ehrsam, C.H.W. Meyer, J.L. Smith, W.L. Tuchman, Message verification and transmission error detection by block chaining, US Patent 4074066, 1976.
[22] J. Kohl, The use of encryption in kerberos for network authentication, in: Proceedings, Crypto '89, Springer-Verlag, 1989.