International Journal of Bifurcation and Chaos, Vol. 26, No. 2 (2016) 1650021 (21 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218127416500218

Fast and Secure Chaos-Based Cryptosystem for Images

Mousa Farajallah* and Safwan El Assad†
Ecole Polytech Nantes-Rue Christian Pauc
44306 Nantes Cedex 3, France
**mousa.farajallah@etu.univ-nantes.fr*
†Safwan.El-Assad@univ-nantes.fr

Olivier Deforges
INSA of Rennes, France
olivier.deforges@insa-rennes.fr

Received October 23, 2014; Revised March 13, 2015

Nonlinear dynamic cryptosystems or chaos-based cryptosystems have been attracting a large amount of research since 1990. The critical aspect of cryptography is to face the growth of communication and to achieve the design of fast and secure cryptosystems. In this paper, we introduce three versions of a chaos-based cryptosystem based on a similar structure of the Zhang and Fridrich cryptosystems. Each version is composed of two layers: a confusion layer and a diffusion layer. The confusion layer is achieved by using a modified 2-D cat map to overcome the fixed-point problem and some other weaknesses, and also to increase the dynamic key space. The 32-bit logistic map is used as a diffusion layer for the first version, which is more robust than using it in 8-bit. In the other versions, the logistic map is replaced by a modified Finite Skew Tent Map (FSTM) for three reasons: to increase the nonlinearity properties of the diffusion layer, to overcome the fixed-point problem, and to increase the dynamic key space. Finally, all versions of the proposed cryptosystem are more resistant against known attacks and faster than Zhang cryptosystems. Moreover, the dynamic key space is much larger than the one used in Zhang cryptosystems. Performance and security analysis prove that the proposed cryptosystems are suitable for securing real-time applications.

Keywords: Dependent permutation; confusion; diffusion; chaos-based cryptosystem; image encryption.

1. Introduction

Today, chaos-based encryption algorithms have been widely used in image and video encryption systems [Hamidouche *et al.*, 2015]. Research has shown that chaos systems are extremely sensitive to the changes of control parameters and initial conditions. They have pseudo-random behavior for nonauthorized parties [Akhshani *et al.*, 2012; El Assad *et al.*, 2014; Kassem *et al.*, 2014; Chiaraluce *et al.*, 2002; Chen *et al.*, 2004; Behnia *et al.*, 2008; Wang *et al.*, 2013; Chang, 2009]. Experimental results show that the chaos-based encryption algorithm can overcome security issues in efficient and adaptive ways compared to the classical encryption ones (such as DES and AES) [Abd El-Latif *et al.*, 2012; Furht & Socek, 2003; Li *et al.*, 2006; Bhargava *et al.*, 2004; Mansour *et al.*, 2012; Bhatnagar & Jonathan Wu, 2012].

Thus, chaos has become a hot topic in the research field over the past decades, and many chaos-based encryption algorithms have been recently introduced [Chen *et al.*, 2004; Socek *et al.*, 2005;

Fridrich, 1997, 1998; Zhang *et al.*, 2005; Masuda *et al.*, 2006; Masuda & Aihara, 2002].

Any cryptosystem must achieve diffusion and confusion effects, in order to be robust and secure against several types of attacks. This has been explained in Shannon's famous paper [Shannon, 1949] "In a strongly ideal cipher all statistics of the cryptogram are independent of the particular key used". Confusion property aims to make the statistical relationship between the cipher image and the secret key as complex and involved as possible, whereas the diffusion property aims to make the statistical relationship between the plain image and the cipher image as complex and involved as possible. The diffusion effect principle can be described as each plaintext byte/bit affects many ciphertext bytes/bits. On the other hand, the confusion principle can be described as: such that the key should not relate to the ciphertext and each bit/byte of the ciphertext should depend on a complex mathematical relation of the key. The former can be achieved using chaotic maps to transfer the single byte/bit effect to other bytes/bits, whereas the latter can be achieved using chaotic maps of permutation and/or substitution. Most chaos-based cryptosystems use chaotic maps to achieve the required diffusion and confusion effects.

Fridrich [1998] proposed a chaos-based encryption scheme based on an iterative pixel permutation and nonlinear processes. The permutation process is achieved using three 2-D chaotic maps: the Backer map, the Cat map, and the Standard map. The nonlinear process is achieved by a nonlinear feedback register.

In this way, Masuda *et al.* [2006] considered two classes of chaotic finite-state maps: key-dependent chaotic S-boxes and chaotic mixing transformation. They proposed two chaotic block ciphers, i.e. uniform and Feistel. In fact, they estimated bounds for differential and linear probability to make their cryptosystems resistant to differential and linear cryptanalysis.

Later, Yang *et al.* [2010], derived a fast image encryption and authentication scheme. A key hash function is introduced to generate a 128-bit hash value from both the plain image and the secret hash keys. The hash value plays the role of secret key for the encryption and the decryption processes, while the secret hash keys are used to authenticate the decrypted image. Permutation and substitution are performed in a single scan of the plain image pixels. The permutation process is achieved by the modified standard map and the substitution process (based on a logistic map) is done in such a way that the change of a particular pixel depends on the accumulated effect of all previous pixel values.

Recently, fast and secure cryptosystems were proposed: Chen *et al.* [2015] have proposed a fast chaos-based image encryption scheme using a dynamic state variables selection mechanism. In the presented algorithm the encryption structure is similar to the structure of Zhang [Zhang *et al.*, 2013], except the process of generating and selecting the dynamic keystream. The security level is reached for one encryption round. Murillo-Escobar *et al.* [2015] have proposed a color image encryption algorithm based on total plain image characteristics, and 1-D logistic map with optimized distribution. They claim that their structure can be implemented in real-time applications. Zhang *et al.* [2013] have proposed a dependent diffusion structure. To the best of our knowledge Zhang cryptosystems seem very secure against attacks (for two encryption rounds in the first cryptosystem and one encryption round in the second cryptosystem) and faster than the previous chaos-based cryptosystems.

The rest of the paper is organized as follows. Direct related works are introduced in Sec. 2. Section 3 presents the general design concept of the proposed cryptosystem and its main differences regarding the Zhang cryptosystems. Sections 4 and 5 describe the detailed mathematical model of the proposed cryptosystems and comparative theoretical security analysis with Fridrich and Zhang cryptosystems. In Sec. 6, security analysis and cryptosystems performance are reported. Finally, the conclusion is presented in Sec. 7.

2. Related Work

In 1997, a chaos-based encryption scheme was introduced by Fridrich [1997, 1998]. It is becoming the core structure of most chaos-based cryptosystems and it has been widely referenced since 1997. The general Fridrich architecture is shown in Fig. 1. The Fridrich encryption scheme is composed of two layers: the first one is the confusion layer using the 2-D Baker chaotic map. This map is used to calculate the new byte position using Eq. (1).

$$B(x', y') = \left(2x, \frac{y}{2}\right) \quad \text{when } 0 \le x < \frac{1}{2}$$

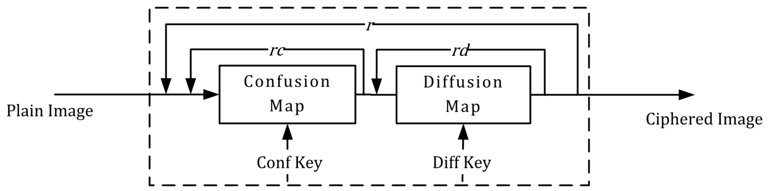


Fig. 1. Fridrich image encryption architecture.

$$B(x', y') = \left(2x - 1, \frac{y}{2} + \frac{1}{2}\right) \quad \text{when } \frac{1}{2} \leq x \leq 1. \tag{1}$$

The second layer is the diffusion one, and is implemented using the following mathematical equations:

$$\begin{aligned} v_k &= v_k + G(v_{k-1}) \text{ Mod } 256 \\ v_{-1} &= \text{initial value.} \end{aligned} \tag{2}$$

The function G is some arbitrary function of the gray level. It was chosen as a fixed random permutation which can be implemented using a simple lookup table.

In [Lian *et al.*, 2005] the authors analyzed the security of the Fridrich scheme. They found some weaknesses, and proposed some improvements to overcome these security failures. In 2010, the Fridrich encryption algorithm was broken by [Solak *et al.*, 2010]. Solak proved that the Fridrich algorithm could be broken using a chosen ciphertext attack. Using this type of attack, some secret permutation of the algorithm has been revealed.

In Zhang's paper [Zhang *et al.*, 2013], two cryptosystems were designed based on the Fridrich architecture. The first one consists of a dependent diffusion layer based on the reverse 2-D cat map. The second algorithm of the Zhang cryptosystem presents new mapping from a pseudo-random position to another pseudo-random one for the confusion effect. Also, a diffusion layer based on the logistic map is used to produce the cipher image.

In these versions, Zhang tried to achieve the confusion and the diffusion effects sequentially. The Fridrich cryptosystem and other cryptosystems are designed so that all pixels are permuted before the pixel values are diffused. However, the Zhang cryptosystems calculate the new location of a pixel and then diffuse that pixel immediately. Thus, the effect of one ciphered pixel is transferred to the next one and so on. From this idea, only two encryption rounds (in the first cryptosystem) and one encryption round (in the second cryptosystem) of the diffusion–confusion process are needed to achieve high security level, instead of many encryption rounds of separated confusion and diffusion processes used in the traditional structures.

3. Proposed Cryptosystems

3.1. *General concepts of the proposed cryptosystems*

The first step of designing a chaos-based encryption algorithm is to define the chaotic maps which will be used in such a structure. These maps are used to achieve the confusion and the diffusion effects, which are the most important properties of any cryptosystem. The general block diagram of all proposed cryptosystems is shown in Fig. 2. The main objective of this structure is to achieve the dependent confusion–diffusion effects byte by byte. All proposed cryptosystems work on the CBC mode and use the El Assad and Noura chaotic generator that produces 32-bit samples [El Assad & Noura, 2011]. Figure 3 shows the general diagram of the

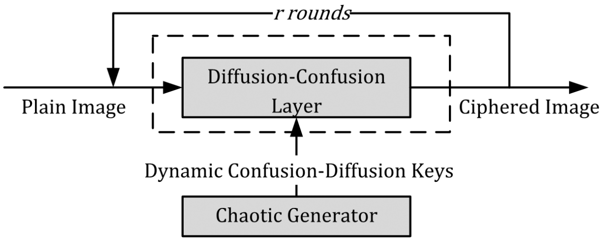


Fig. 2. General block diagram of the proposed cryptosystems.

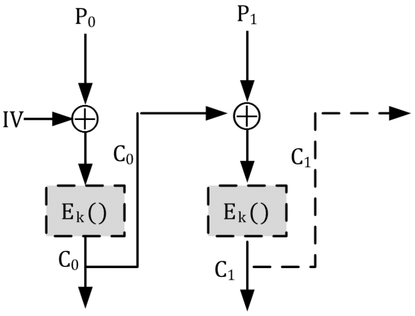


Fig. 3. Encryption structure of the CBC mode.

encryption part of the CBC mode. In Fig. 3, P_0 is the first block from the plain image, IV is the initial vector generated by the chaotic generator, C_0 is the first encrypted block which will be transferred to the receiver side. The dash boxes represent the encryption/decryption parts of the proposed cryptosystem (see Figs. 2, 4 and 7).

3.2. *General differences of the proposed cryptosystem with the Zhang one*

In this section, we introduce some general differences between our proposed cryptosystem of the three versions and the Zhang one:

- In the Zhang cryptosystem, a simple implementation of the logistic map was used to generate and manage the dynamic keys. In all versions of our proposed cryptosystem, a strong and robustness tested chaotic generator [El Assad & Noura, 2011] is used to manage and generate the dynamic keys. These dynamic keys are used for the dependent confusion–diffusion layer.
- To overcome the fixed-point problem of the standard 2-D cat map, the Zhang cryptosystem selects a random pixel arr(r_x^j, r_y^j) and swaps it with the first pixel in the plain image arr$(0, 0)$. In our proposed cryptosystem, this problem is solved by using the modified 2-D cat map. This modification of the standard 2-D cat map also enhances the security of the proposed cryptosystem. Indeed the number of dynamic keys in the 2-D cat map increases from two dynamic keys to four. The first pixel $(0, 0)$ can be mapped to any new position (i_n, j_n).
- The Zhang cryptosystem works on the whole image. It is well known that the image encryption algorithms that work on the whole image give bad results for error propagation. The influence of one error bit of the ciphered data (due to the channel) on the decryption algorithm depends on the cryptographic modes. All versions of our proposed cryptosystem perform the encryption/decryption operations based on the Cipher-Block Chaining (CBC) mode [Ehrsam *et al.*, 1978].
- One of the most important differences between our proposed cryptosystems and the Zhang cryptosystem is in the structure of the dynamic keys. In the Zhang cryptosystem, the dynamic keys consist of two keys (q_i, p_i) for the reverse 2-D cat map. These key values are changed just twice for the whole encryption process. For the logistic map, t is the initial value. The value of t is changed in each byte. When comparing with our proposed cryptosystem, in the 2-D cat map, four keys are used $(v, u, ri$, and $rj)$. These keys' values are changed for every new encryption round and also for every new block.
- All chaos-based cryptosystems that use more than one encryption round (i.e. $r > 1$) to reach the required security level, must save their dynamic keys for all rounds in order to use them later in the decryption process. From a security point of view, it is normal that the security level of any cryptosystem is strongly related to the environment where these dynamic keys have been temporarily saved. To the best of our knowledge, the only possible method to manage the dynamic keys in the decryption process in case of more than one decryption round is to save them temporarily, since the decryption process is achieved by starting from the last decryption round and finishing with the first decryption round. It is important to note that the chaotic generator has to be a noninvertible generator. To obtain the current key, the chaotic generator should generate all previous keys and use them in the reverse order.

All previously mentioned points are taken into consideration in the design process of a fast and secure cryptosystem, and it should use only one encryption round $(r = 1)$, to obtain the required security level.

4. First Proposed Cryptosystem

The first proposed cryptosystem has some similarity to the Zhang cryptosystem [Zhang *et al.*, 2013]. In this cryptosystem, the two dependent confusion–diffusion layers are the modified 2-D cat map (used to achieve the confusion effect) followed by the discrete logistic map (to achieve the diffusion effect). Using this structure, the required confusion–diffusion effects are obtained by one encryption round, and hence the execution time is decreased.

4.1. *Encryption scheme of the first proposed cryptosystem*

In the encryption scheme (see Fig. 4), for the first block, each pixel from the plain block $(p_0(k))$ is XOR-ed with the initial byte $(iv(k))$ from the initial

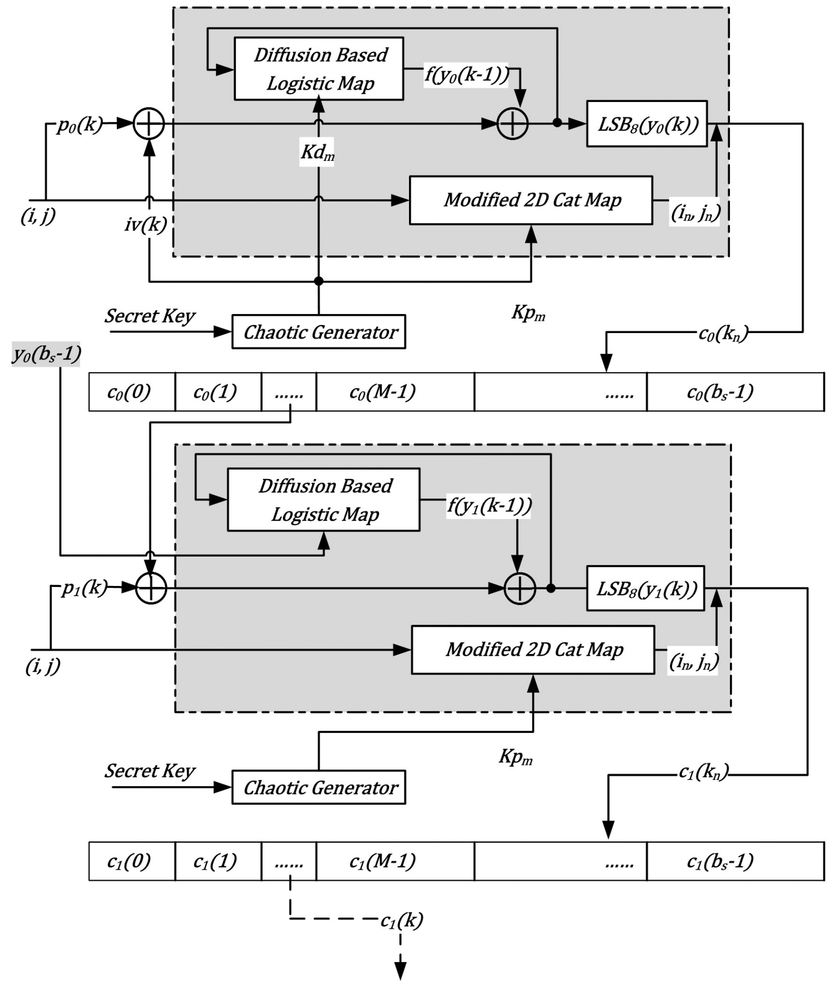


Fig. 4. Encryption structure of the first proposed cryptosystem.

vector (IV), then the output is XOR-ed with the discrete logistic map output to carry out the diffusion process. Then, the eight least significant bits resulting from the diffusion process $\mathrm{LSB}_8(y_0(k))$ are relocated using the modified 2-D cat map to obtain the ciphered pixel at the new position ($c_0(k_n)$) [Farajallah *et al.*, 2013a; Farajallah *et al.*, 2013b]. It is important to note that the input of the discrete logistic map is based on the previous ciphered pixel (since $c_0(k_n) = \mathrm{LSB}_8(y_0(k))$ and the input of the discrete logistic map is 32 bits and the ciphered pixel is eight bits. That is why the cryptosystem takes ($y_0(k-1)$) before the LSB_8 function and not after. For the first encrypted byte, the input of the discrete logistic map is Kd_m, and this value is reinitialized every new encryption round. Because the $c_0(k_n)$ is only a part of the logistic map input, it is impossible to recover $y_0(k-1)$ from $c_0(k_n)$ only. The encryption of the next blocks is almost the same. Each pixel from the plain block ($p_l(k)$) is XOR-ed with ciphered byte from the previous block at the same position (i.e. $c_{l-1}(k)$ to achieve the CBC mode). Then the rest of the operations are the same as in the first encryption block. The 2-D cat map was tested and analyzed by [Fridrich, 1998] and [Wong *et al.*, 2008]. To overcome the fixed-point problem of the Arnold cat map model, the parameters ri and rj are added to the standard model. Also, in our scheme the elements of the square matrix and the parameters ri and rj of Eq. (3) become dynamic, they form the dynamic keys of the permutation process.

$$\begin{bmatrix} i_n \\ j_n \end{bmatrix} = \mathrm{Mod}\left(\begin{bmatrix} 1 & u \\ v & 1+uv \end{bmatrix}\begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} ri+rj \\ rj \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix}\right). \tag{3}$$

Equation (3) is a one-to-one function, which means that each point of the square matrix can be transferred to exactly one unique point. So, instead of exchanging the values at the new position (i_n, j_n) with the old one (i, j), we use a transfer operation because of its speed compared to the swap operation that is usually used. The block size b_s is M^2 (M is the square root of the block size in our proposed cryptosystem). The system parameters u, v, ri and rj are in the range of $[0, M-1]$. The structure of the dynamic keys which are produced by the chaotic generator during the permutation process is:

$$\begin{aligned} Kp &= [Kp_0 \| Kp_1 \| Kp_2 \| \cdots \| Kp_{r-1}] \\ Kp_m &= [u_m \| v_m \| ri_m \| rj_m]. \end{aligned} \tag{4}$$

The modulo operation of Eq. (3) makes it a non-invertible equation. But it is still a reversible one. Thus, in the decryption part of the proposed cryptosystem, the reverse layer is also achieved by Eq. (3).

The implementation of this modified 2-D cat map is carried out by an optimized process, indeed:

- the value of $Z_1 = ri + rj$ is calculated once per round.
- the value of $Z_2 = u \times v + 1$ is calculated once per round.
- the value of $Z_3 = v \times i + rj$ is calculated M times per round.

Then the modified 2-D cat map is implemented as:

$$\begin{bmatrix} i_n \\ j_n \end{bmatrix} = \text{Mod}\left(\begin{bmatrix} i + u \times j + Z_1 \\ Z_3 + Z_2 \times j \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix} \right). \tag{5}$$

The Logistic map is a nonlinear chaotic discrete function that produces random sequences. In the proposed cryptosystem, the logistic map is used as a diffusion function to achieve the diffusion effect, by transferring the effect from one byte in the block to other bytes in the same block. This structure makes the proposed cryptosystem highly sensitive to the plaintext. The mathematical model of the discrete logistic map is:

$$X_{k+1} = \begin{cases} \left\lfloor \dfrac{X_k \times (2^N - X_k)}{2^{N-2}} \right\rfloor \\ \qquad \text{if } X_k \neq [3 \times 2^{N-2}, 2^N] \\ 2^N - 1 \quad \text{if } X_k = [3 \times 2^{N-2}, 2^N] \end{cases} \tag{6}$$

where X_{k+1} is the new value calculated from the previous one X_k. N is the number of bits representing the integer output of the discrete logistic map, which is equal to 32 bits in all versions of our proposed cryptosystem. Figure 4 shows the block diagram of the encryption part of the first version of the proposed cryptosystem. From the figure, we write the encryption mathematical model as:

$$c_l(k_n) = \text{LSB}_8[y_l(k)] \tag{7}$$

$$y_l(k) = p_l(k) \oplus s_{l-1}(k) \oplus f(y_l(k-1)) \tag{8}$$

where $y_l(k)$ is a 32-bit variable, $p_l(k)$, $s_{l-1}(k)$ are 8-bit variables and f is the logistic map. The following remarks should be considered:

(1) During the encryption, Eq. (8) should be evaluated before Eq. (7), for each byte of a block and for all blocks.
(2) The input of the logistic map for $k = 0$ is kd_m when $l = 0$ and it is $y_{l-1}(b_s - 1)$ for $l > 0$.
(3) For $k > 0$ and for all l, the input of the logistic map is the result of Eq. (8) and not the previous output [see Eq. (6)].

Note that:

$$k = i \times M + j, \quad k_n = i_n \times M + j_n$$

i_n and j_n are calculated using Eq. (3). The sequence $s_{l-1}(k)$ is given by the following equation:

$$s_{l-1}(k) = \begin{cases} iv(k) & \text{if } l = 0 \\ c_{l-1}(k) & \text{if } l > 0 \end{cases} \tag{9}$$

where

$$l = 0, 1, 2, \ldots, b_n - 1, \quad k = 0, 1, 2, \ldots, b_s - 1$$

$$IV = \{iv(0), iv(1), iv(2), \ldots, iv(b_s - 1)\}$$

b_s is block size in bytes

$$\begin{aligned} b_n &= \frac{\text{image size}}{\text{block size}} \\ &= \frac{L \times C \times P}{b_s}, \text{ is the number of blocks} \end{aligned}$$

where, L, C, and P are the number of lines, the number of columns, and the number of planes of the image respectively.

In Algorithm 1, we describe in detail the exact steps to achieve the ciphering process for all blocks.

4.2. *Proposed chaotic generator*

The generator used in this cryptosystem is a simplified implementation of the one published by El Assad and Noura in a patent [El Assad & Noura, 2011]. It consists of two chaotic maps (i.e. Skew

Algorithm 1 Encryption steps

```
Generate the $IV$ values using the chaotic generator to encrypt the first block of the image.
for $m = 0$: $r - 1$: $step = 1$ do
  Generate the values of $Kd_m$ and $Kp_m$ using the chaotic generator to encrypt the first block.
  $k = 0$
  for $i = 0$: $M - 1$: $step = 1$ do
    for $j = 0$: $M - 1$: $step = 1$ do
      Initialize the value of $s_{-1}(k) = iv(k)$
      Calculate $(i_n,\ j_n)$ using equation (3)
      Calculate $k_n = i_n \times M + j_n$
      Calculate $y_0(k)$ value using equation (6) and equation (8)
      Calculate $c_0(k_n)$ value using equation (7)
      $k = k + 1$
    End $j$
  End $i$
End $m$
for $l = 1$: $b_n - 1$: $step = 1$ do
  for $m = 0$: $r - 1$: $step = 1$ do
  Generate the values of $Kp_m$ using the chaotic generator to encrypt the current block of the plain
   image.
  $k = 0$
    for $i = 0$: $M - 1$: $step = 1$ do
      for $j = 0$: $M - 1$: $step = 1$ do
        Initialize the value of $s_{l-1}(k) = c_{l-1}(k)$
        Calculate $(i_n,\ j_n)$ using equation (3)
        Calculate $k_n = i_n \times M + j_n$
        Calculate $y_l(k)$ value using equation (6) and equation (8)
        Calculate $c_l(k_n)$ value using equation (7)
        $k = k + 1$
      End $j$
     End $i$
  End $m$
End $l$
```

Tent and Piecewise Linear Chaotic Map), connected in parallel as shown in Fig. 5. Each map is perturbed using a linear feedback shift register (LFSR). This ensures a very large periodicity for all generated sequences. The generator produces 32-bit (four bytes) samples. For our first version of the proposed cryptosystem, the following generated samples are needed to encrypt an image:

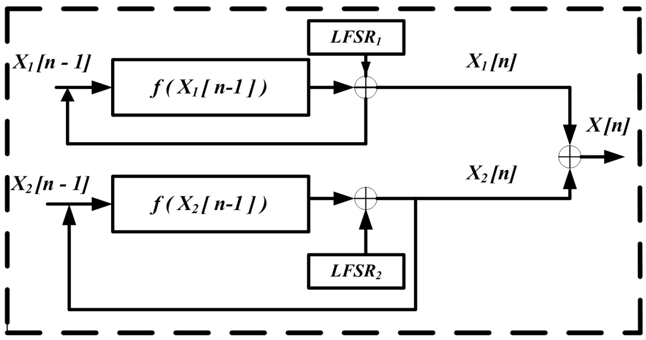


Fig. 5. Proposed chaotic sequence generator.

Kd is a 32-bit size, supplied for the first block, and changed for every encryption round, so the total number of required samples for Kd is:

$$Kd_{\text{samples}} = r. \tag{10}$$

IV is a vector of b_s bytes, it is supplied one time for the first block, so the total number of required samples for IV is:

$$IV_{\text{samples}} = \frac{b_s}{4}. \tag{11}$$

Kp consists of four variables (u, v, ri and rj), each variable is q bits with $q = \log_2 \sqrt{b_s}$, so the total number of required samples for Kp is:

$$Kp_{\text{samples}} = \left\lceil \frac{b_n \times r \times 4 \times \log_2 \sqrt{b_s}}{32} \right\rceil. \tag{12}$$

From Eqs. (10)–(12), the total samples required to encrypt an image is:

$$\text{Total}_{\text{samples}} = Kd_{\text{samples}} + IV_{\text{samples}} + Kp_{\text{samples}}. \tag{13}$$

The discrete Skew Tent Map and the discrete PWLCM are defined in the following equations.

First, the discrete Skew Tent Map is defined as [Masuda *et al.*, 2006]:

$$X[n] = F[X[n-1]] = \begin{cases} \left\lfloor \dfrac{2^N \times X[n-1]}{P} \right\rfloor & \text{if } 0 < X[n-1] < P \\ 2^N - 1 & \text{if } X[n-1] = P \\ \left\lfloor \dfrac{2^N \times (2^N - X[n-1])}{2^N - P} \right\rfloor & \text{if } P < X[n-1] < 2^N \end{cases} \tag{14}$$

where P is the control parameter, ranging from 1 to $2^N - 1$, and $N = 32$ bits, is the finite precision.

Second, the PWLCM map is defined as [Lian *et al.*, 2007]:

$$X[n] = F[X[n-1]] = \begin{cases} \left\lfloor \dfrac{2^N \times X[n-1]}{P} \right\rfloor & \text{if } 0 < X[n-1] < P \\ \left\lfloor \dfrac{2^N \times (X[n-1] - P)}{2^{N-1} - P} \right\rfloor & \text{if } P < X[n-1] < 2^{N-1} \\ \left\lfloor \dfrac{2^N \times (2^N - X[n-1] - P)}{2^{N-1} - P} \right\rfloor & \text{if } 2^{N-1} \le X[n-1] < 2^N - P \\ \left\lfloor \dfrac{2^N \times (2^N - X[n-1])}{P} \right\rfloor & \text{if } 2^N - P \le X[n-1] < 2^N - 1 \\ 2^N - 1 & \text{otherwise.} \end{cases} \tag{15}$$

The control parameter P of the PWLCM ranges from 1 to $2^{(N-1)} - 1$. The proposed chaotic generator has the following cryptographic properties: pseudo-random mapping, delta-like auto-correlation, nearly zero cross-correlation, uniform distribution, passing empirical statistical test NIST 800-22 (National Institute of Standards and Technology) tests [Rukhin *et al.*, 2001], and having a large size of the secret key. The size of the secret key is determined by four initial conditions: two values for the Skew Tent and PWLCM maps of size N and the others for the two LFSRs, and two control parameters, i.e. P_1 (for the Skew Tent) and P_2 (for the PWLCM).

$$\begin{aligned} |K| &= 2 \times N + |P_1| + |P_2| + |K_1| + |K_2| \\ &= 169 \text{ bits.} \end{aligned} \tag{16}$$

Moreover, the implemented generator is very secure against known attacks: ciphertext attack and chosen-plaintext attack (this latter attack is the easiest one among all known attacks). The objective of these attacks is to determine the secret key that was used. The ciphertext attack, means here, guessing the secret key from the generated sequences. This attack is ineffective, because the generated sequences are obtained by a combination of sequences supplied by two different nonlinear maps. And it is impossible to analyze sequences of each map separately. The chosen-plaintext attack is equivalent here to the key sensitivity attack, because the only input of the system is the secret key. From the main property of any chaotic system (the extreme sensitivity to even one bit change of the secret key), the implemented generator passes this test (see also, the test done in Sec. 6.3 about the key sensitivity attack).

4.3. *Decryption scheme of the first proposed cryptosystem*

The decryption scheme of the first proposed cryptosystem is almost identical to the encryption one. Figure 6 shows the decryption structure of the CBC mode in all proposed cryptosystems, while Fig. 7 shows the decryption structure of the first proposed cryptosystem.

The decryption equation resulting from the encryption equations is: (7) and (8):

$$p_l(k) = c_l(k_n) \oplus s_{l-1}(k) \oplus \mathrm{LSB}_8[f(y_l(k-1))] \tag{17}$$

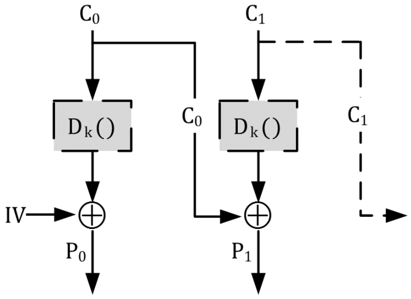


Fig. 6. Decryption structure of the CBC mode.

where

$$y_l(k) = p_l(k) \oplus s_{l-1}(k) \oplus f(y_l(k-1)). \tag{18}$$

As we can see, Eq. (17) is firstly evaluated followed by Eq. (18) for each byte of a block and for all blocks.

Algorithm 2 Decryption steps

```
Generate $Kd_m$, $Kp_m$ for all $m$ values (i.e. $m = 0, 1, \ldots, r-1$) and $IV$ using the chaotic generator to
decrypt the first ciphered block.
for $m = r - 1$: 0: $step = -1$ do
  $k = 0$
  for $i = 0$: $M - 1$: $step = 1$ do
    for $j = 0$: $M - 1$: $step = 1$ do
      Initialize the value of $s_{-1}(k) = iv(k)$
      Calculate $(i_n,\ j_n)$ using equation (3)
      Calculate $k_n = i_n \times M + j_n$
      Calculate $p_0(k)$ value using equation (17)
      Calculate $y_0(k)$ value using equation (6) and equation (8)
      $k = k + 1$
    End $j$
  End $i$
End $m$
for $l = 1$: $b_n - 1$: $step = 1$ do
 Generate the $Kp_m$ for all $m$ values (i.e. $m = 0, 1, \ldots, r-1$)
 using the chaotic generator to decrypt the current block from the ciphered image.
 for $m = r - 1$: 0: $step = -1$ do
  $k = 0$
  for $i = 0$: $M - 1$: $step = 1$ do
    for $j = 0$: $M - 1$: $step = 1$ do
      Initialize the value of $s_{l-1} = c_{l-1}k$
      Calculate $(i_n,\ j_n)$ using equation (3)
      Calculate $k_n = i_n \times M + j_n$
      Calculate $p_l(k)$ value using equation (17)
      Calculate $y_l(k)$ value using equation (6) and equation (8)
      $k = k + 1$
    End $j$
  End $i$
 End $m$
End $l$
```

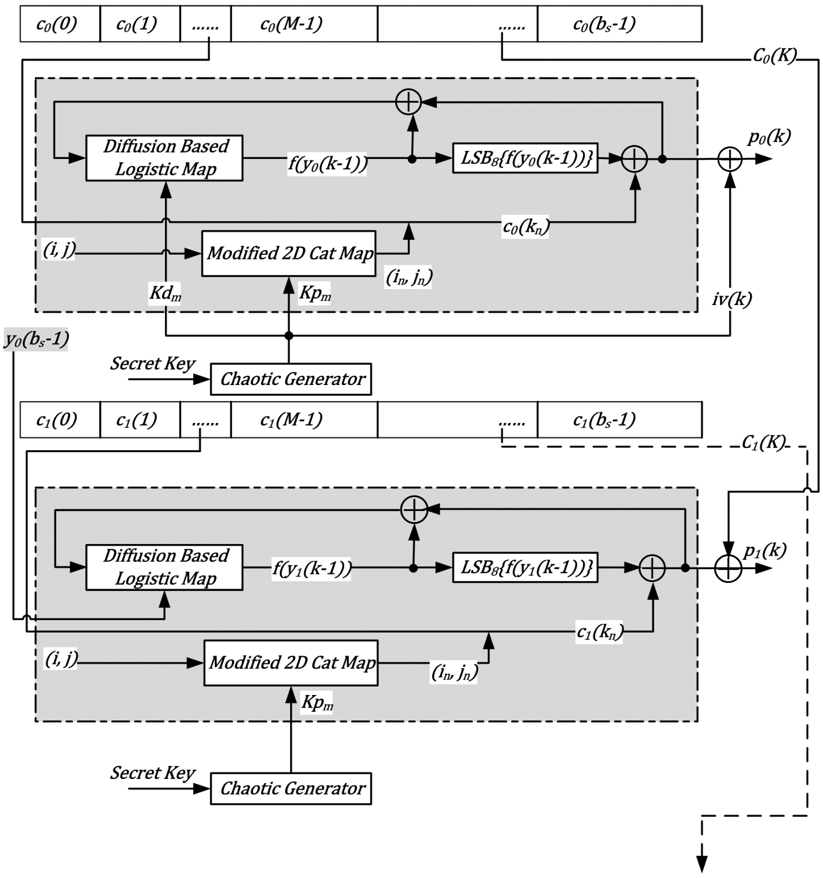


Fig. 7. Decryption structure of the first proposed cryptosystem.

In reality, as we can see in Fig. 7, Eq. (18) is implemented as follows:

$$y_l(k) = c_l(k_n) \oplus s_{l-1}(k) \oplus \mathrm{LSB}_8[f(y_l(k-1))]$$
$$\oplus\, s_{l-1}(k) \oplus f(y_l(k-1))$$
$$y_l(k) = c_l(k_n) \oplus \mathrm{LSB}_8[f(y_l(k-1))] \oplus f(y_l(k-1)). \tag{19}$$

4.4. *Analysis of the first proposed cryptosystem*

Lian *et al.*, in their paper [2005], analyzed the Fridrich model, and they pointed out some possible attacks on that model. We apply and analyze these attacks on our proposed cryptosystem to ensure its robustness against them.

4.4.1. *Dynamic key space analysis of Fridrich, Zhang and our cryptosystems*

Lian in his paper assumes that the dynamic key space of the whole Fridrich cryptosystem is $S^r = (S_1 \times S_2)^r$, where S_1 is the dynamic key space of the confusion layer, S_2 is the dynamic key space of the diffusion layer, and r is the number of iterations.

The total dynamic key space of the Fridrich model can be calculated as:

$$S_1 = (N^2)^r, \quad S_2 = L^r$$
$$KS_{\text{Fridrich}} = (N^2)^r \times L^r = (N^2 \times L)^r$$

where M is the square root of the tested image size and $L = 256$, is the number of gray levels.

In the Zhang cryptosystem, the first algorithm has q_1, q_2, p_1 and p_2 in the range [0, 511], and also t_0 which is eight bits, so:

$$KS_{\text{Zhang1}} = (N^4 \times 2^8)^r.$$

The second algorithm has q_1, q_2, p_1 and p_2 in the range [0, 511], and also temp1 and temp2 of eight bits each, so:

$$KS_{\text{Zhang2}} = (N^4 \times 2^{16})^r.$$

In our proposed cryptosystem, the total dynamic key space is defined as:

$$S^r = (S_1 \times S_2)^r$$

where S_1 is the dynamic key space of the confusion layer (the modified 2-D cat map), S_2 is the dynamic key space of the diffusion layer (the logistic map).

The total dynamic key space of our proposed cryptosystem can be calculated as:

(1) The whole image is divided into a number of blocks and the dynamic keys are changed for every new encryption round and every new block.
(2) Dynamic keys for the confusion layer are (u, v, ri and rj: in the range of $[0, M-1]$). Then, the confusion key space is M^4.
(3) The diffusion key space (S_1) is 32 bit instead of eight bit as in the Fridrich or the Zhang cryptosystems.
(4) The total number of blocks is b_n where $b_n = \frac{L\times C\times P}{b_s} = \frac{L\times C\times P}{M^2}$.

$$KS = (S_1 \times S_2)^r \times b_n$$

$$S_1 = M^4, \quad S_2 = 2^{32}.$$

For $r = 1$:

$$KS_{\text{Proposed}} = (M^4 \times 2^{32}) \times \frac{L\times C\times P}{M^2}.$$

As an example and to make the comparison between cryptosystems, of Zhang, Fridrich and ours, the gray-scale Lena image of 512 × 512 is taken, and then for one encryption round (remark, for our proposed cryptosystem $M = 32$):

$KS_{\text{Fridrich}} = 512^2 \times 256 = 2^{26}$

$KS_{\text{Zhang1}} = 512^4 \times 2^8 = 2^{44}$

$KS_{\text{Zhang2}} = 512^4 \times 2^{16} = 2^{52}$

$KS_{\text{Proposed}} = (32^4 \times 2^{32}) \times (\frac{512\times 512}{32^2}) = 2^{60}$.

It is clear from the previous calculations that the dynamic key space of our proposed cryptosystem is $\mathbf{2^{16}}$ times more than the first Zhang cryptosystem, and $\mathbf{2^{8}}$ times more than the second Zhang cryptosystem.

4.4.2. *Chosen-plaintext attack*

Fridrich proved that his proposed model is secure against a chosen-plaintext attack based on the fact that the difference between the ciphertexts encrypted by the same key for two plaintexts differing on one bit is large enough to keep a high security level against the chosen-plaintext attack. However, Lian *et al.* [2005] pointed out another kind of attack that can be used to cryptanalyze the Fridrich model. Since the fixed-point problem was not solved in the 2-D cat map, Baker or standard maps were used in the Fridrich model, and so the cipher of the first plain pixel of any image will remain in the first position (that means c_0 is the encryption of the p_0, and so, no permutation is done on the first pixel). Then it is easy to find the initial value of the diffusion key (Q_{-1}) in the Fridrich model (more details of this attack can be found in [Lian *et al.*, 2005]).

In Zhang cryptosystems, a simple solution to overcome this problem is introduced, the solution is carried out by swapping the first pixel with a random pixel from the image before starting the encryption process.

In our proposed cryptosystem, the first kind of chosen-plaintext attack is solved, and it satisfies a high security level. This is well stated by our proposed cryptosystem in Sec. 6.2.

The fixed-point problem does not exist in our proposed cryptosystem, since it is solved by adding the ri and the rj dynamic keys to the original 2-D cat map, and the chaotic generator insures that ri and rj will never be zero at the same time.

Our solution has two advantages, first, it increases the dynamic key space of the 2-D cat map, second, any pixel can be mapped to any position with the same probability.

4.4.3. *Some specific differences in the diffusion process*

Zhang cryptosystems implement the logistic map based on eight bits. This means, that the input and the output of the logistic map for all operations are in eight bits, whereas in our proposed cryptosystem, the logistic map is implemented to receive 32 bits as input and to produce 32 bits as output. As a result, the dynamic key of the logistic map is increased from eight bits to 32 bits, and the security level of the overall cryptosystem is increased. Finally, it is well known that the implementation of the logistic map based on eight bits has some security weaknesses and failures.

Finally, as the proposed cryptosystem uses new dynamic keys for each new block, then, the cryptosystem is secure against the chosen-plaintext attacks, according to the fact that a chosen-plaintext attack will be useless if different keys are used to encrypt different plaintexts in the same message.

5. Second Proposed Cryptosystem

The construction and design process of the following cryptosystems begins by keeping in mind the structure of the first proposed cryptosystem. As we know, there is always a trade off between the security level and the encryption time. Increasing the security level in general leads to making the system more complex, and then adding some additional delays on the encryption operations. For this reason, two subversions of the second version are described in this section. These subversions have the same structure as the first proposed cryptosystem, but using a different diffusion layer, namely the Finite Skew Tent Map (FSTM) as a generator instead of the logistic map.

5.1. *Finite Skew Tent Map as diffusion layer*

The diffusion layer is implemented using a modified version as a generator of the original FSTM in [Masuda *et al.*, 2006; Masuda & Aihara, 2002], based on a lookup table of eight bits for the first subversion and on a mathematical calculation of 32 bits for the second subversion [see Eq. (20)]. The FSTM has a better nonlinear transformation than the logistic map and so its diffusion is stronger than that of the logistic map. This implies that the cryptosystem is more resistant against differential cryptanalysis attacks. So, using the FSTM as a dependent diffusion layer increases cryptosystem sensitivity to plain sensitivity attacks. The mathematical model of the modified FSTM generator is [Farajallah *et al.*, 2013b]:

$$F(X) = \begin{cases} \left\lfloor \dfrac{Q}{A_m} \times X \right\rfloor + A_{0m} \text{ Mod } Q \\ \quad 0 \le X \le A_m \\ \left\lfloor \dfrac{Q \times (Q - X)}{Q - A_m} \right\rfloor + 1 + A_{0m} \text{ Mod } Q \\ \quad A_m < X < Q \end{cases} \tag{20}$$

where

$$A_{0m}, X, F(X) \in \{0, 1, 2, \ldots, Q-1\} \quad \text{and}$$

$$A_m \in \{1, 2, \ldots, Q-1\}.$$

In Eq. (20), Q is equal to (2^8) for the first subversion (lookup table), and equal to (2^{32}) for the second subversion (the mathematical implementation of the FSTM). Relative to the first proposed cryptosystem, the input value X of Eq. (20) is Eq. (8), while $F(X)$ in Eq. (20) is $f(y_l(k-1))$ and the initial value X_0 is Kd_m (see Figs. 4 and 7).

The structure of the dynamic keys during the diffusion process is:

$$\begin{aligned} Ks &= \lfloor Ks_0 \| Ks_1 \| Ks_2 \| \cdots \| Ks_{r-1} \rfloor \\ Ks_m &= A_m \| A_{0m} \end{aligned} \tag{21}$$

where r is the number of rounds for each block. In the standard FSTM, the fixed-point problem is not solved (i.e. when the input of the FSTM is ZERO the output is ZERO). To overcome this problem we introduce A_{0m} in the FSTM equation. As a result, any input value is mapped to any output value with the same probability without any restrictions. Moreover, introducing the dynamic key A_{0m} increases the dynamic key space.

The first subversion uses eight bits to implement the FSTM generator as a lookup table, and so it is faster than the first cryptosystem, while still having a high security level. The lookup table is created since the input and the output of the FSTM are limited to eight bits. Figure 4 can be used to describe the encryption process of this subversion. The first step is to generate the dynamic keys (Kp_m, X_0, A_m and A_{0m}). The permutation process is applied on the plain pixels by taking each byte, and calculating their new position according to Eq. (3). Then, Eq. (8) is applied to obtain the y_k value which defines the ciphered pixel as shown in Eq. (7). Note that the value of y_k is eight bits and so there is no need for the LSB function of Eq. (7). It is important to note that Eq. (20) is implemented in lookup table of 64 KB size without the A_{0m}, the returned value from the lookup table is added to the A_{0m}.

The second subversion uses 32 bits to implement the FSTM as a generator. It is slower than the first cryptosystem, but it has a dynamic key space greater than the first cryptosystem and thus it is very robust against cryptanalysis. The encryption process in this version is exactly identical to the first one, except in this version Eq. (20) is used instead of Eq. (6).

Using the lookup table based on eight bits for the first subversion, the first eight bits from each sample of the chaotic generator are taken to be used as the dynamic key (A_m or A_{0m}), and the remaining 24 bits are skipped. If the first eight bits are zeros, then the next eight bits are taken and so on.

The dynamic keys of the first subversion need two samples for each round of each block. It is important to note that the used chaotic generator never produces a sample of 32 bits where all of the bits are zeros. The dynamic keys A_m and A_{0m} in the second subversion are 32 bits each, so, two samples are also needed for each round of each block.

$$Ks_{\text{samples}} = 2 \times r. \tag{22}$$

So the total number of required samples in this version is:

$$\text{Total}_{\text{samples}} = 2 \times r + \frac{b_s}{4} + \left\lceil \frac{b_n \times r \times 4 \times \log_2 \sqrt{b_s}}{32} \right\rceil. \tag{23}$$

5.2. *Analysis of the second proposed cryptosystem*

In this section, we analyze the dynamic key space and the chosen-plaintext attacks.

5.2.1. *Dynamic key space analysis*

In our proposed cryptosystem, the total dynamic key space is:

$$KS = (S_1 \times S_2)^r \times b_n.$$

For one encryption round ($r = 1$):

First subversion key space:

$$S_1 = M^4$$

$S_2 = 2^{24}$, because X_0, A_m and A_0m are eight bits each.

$$KS_{\text{Subversion1}} = M^4 \times 2^{24} \times \frac{L \times C \times P}{M^2}.$$

Second subversion key space:

$$S_1 = M^4$$

$S_2 = 2^{96}$, because X_0, A_m and A_0m are 32 bits each.

$$KS_{\text{Subversion2}} = M^4 \times 2^{96} \times \frac{L \times C \times P}{M^2}.$$

Again, to make the same comparison between our proposed cryptosystem, the Zhang and the Fridrich cryptosystems, the Lena image of 512 × 512 bytes is taken, then for one encryption round (remark, for our proposed cryptosystem $M = 32$):

$KS_{\text{Fridrich}} = 2^{26}$
$KS_{\text{Zhang1}} = 2^{44}$
$KS_{\text{Zhang2}} = 2^{52}$
$KS_{\text{Subversion1}} = (32^4 \times 2^{24}) \times \frac{512 \times 512}{32^2} = 2^{52}$
$KS_{\text{Subversion2}} = (32^4 \times 2^{96}) \times \frac{512 \times 512}{32^2} = 2^{124}.$

It is clear from the previous calculations that the dynamic key space of the first subversion is $\mathbf{2^8}$ times more than the first Zhang cryptosystem, and the same as the second Zhang cryptosystem, whereas the dynamic key space of the second subversion is $\mathbf{2^{80}}$ times more than the first Zhang cryptosystem, and $\mathbf{2^{72}}$ times more than the second Zhang cryptosystem.

5.2.2. *Chosen-plaintext attack*

In this cryptosystem version, the modified 2-D cat map is the same as before, and so the analysis of the chosen-plaintext attack is the same. The FSTM is enhanced by adding the parameter A_0. This means that the fixed-point problem is solved also for the diffusion layer, and so, this type of attack will be useless.

6. Performance and Security Analysis

The performances of the proposed cryptosystems have been evaluated by: measuring the encryption/decryption speed, the throughput, and the number of cycles for each algorithm. The obtained results are compared with results of other known cryptosystems. Experimental and statistical analysis is used to evaluate the security of the proposed cryptosystem for all kinds of known attacks in the literature.

6.1. *Time and complexity analysis*

The speed evaluation of our proposed cryptosystem is carried out using a C compiler, on a PC with 3.1 GHz processor Intel Core i3-2100 CPU, 4 GB RAM, and Windows 7, 32-bit Operation System. It encrypts different images of different sizes. (256 × 256 × 3, 512 × 512 × 3 and 1024 × 1024 × 3). We compare the speed of our proposed cryptosystems with the fastest chaos-based cryptosystems. In particular, the security and the performance analysis of the proposed cryptosystems are compared with Zhang [Zhang *et al.*, 2013] cryptosystem, since, to the best of our knowledge, it is the

Table 1. Average encryption/decryption time of the proposed algorithm (in milli-seconds).

Our Cryptosystem Version	bs	256 × 256 × 3	512 × 512 × 3	1024 × 1024 × 3
Proposed V1	256	2.21/2.82	8.75/11.16	34.87/44.34
Proposed V1	1024	2.04/2.68	8.08/10.57	31.85/41.83
Proposed V2-8 bit	256	1.73/1.78	6.82/7.10	27.01/28.04
Proposed V2-8 bit	1024	1.38/1.45	5.42/5.74	21.17/22.39
Proposed V2-32 bit	256	4.57/5.24	18.29/20.89	73.05/83.43
Proposed V2-32 bit	1024	4.15/4.79	16.56/19.08	66.12/76.17

fastest chaos-based cryptosystem. Table 1 presents the encryption and the decryption times for our proposed algorithms, based on two different block sizes ($b_s = 256$ and $b_s = 1024$ bytes) and the image under test was Lena. The time calculation process of encryption and decryption is evaluated as follows: the test image (Lena with block size 1024) is encrypted for 1000 different secret keys, then the average of these executions is calculated. From Table 2, it is clear that our proposed cryptosystems are faster than both of the Zhang algorithms and other known cryptosystems. Table 3 presents a comparison of performance of our proposed cryptosystem with some known recent cryptosystems in the literature. The performance is evaluated in terms of encryption throughput (running speed) in Mega Byte Per Second (MBps) and number of needed cycles to encrypt/decrypt one byte. The encryption throughput is calculated by Eq. (24) in bytes, whereas Eq. (25) is used to compute the number of cycles that are needed to encrypt one byte.

$$ET = \frac{\text{Image}_{\text{Size}}(\text{Byte})}{\text{Encryption}_{\text{Time}}(\text{second})} \tag{24}$$

$$\text{Number of cycles per Byte} = \frac{\text{CPU Speed}_{(\text{Hertz})}}{ET_{(\text{Byte})}}. \tag{25}$$

It is clear from Tables 1–3 that our proposed cryptosystem in all proposed versions is faster than the other chaos-based cryptosystems. The security

Table 2. Encryption/decryption time of different algorithms (in milli-seconds).

Proposed Cryptosystem	256 × 256 × 3	512 × 512 × 3	1024 × 1024 × 3
Proposed V1	2.04/2.68	8.08/10.57	31.85/41.83
Proposed V2-8 bit	1.38/1.45	5.42/5.74	21.17/22.39
Proposed V2-32 bit	4.15/4.79	16.56/19.08	66.12/76.17
Zhang 1 [Zhang *et al.*, 2013]	7.5/7.5	30/30	120/120
Zhang 2 [Zhang *et al.*, 2013]	7.5/8.25	30/33	120/132
Wang [Wang *et al.*, 2011]	7.79/8.39	31.16/33.54	124.64/134.16
Akhshani [Akhshani *et al.*, 2012]	14.4	57.6	230.4
Wong [Wong *et al.*, 2008]	15.59/16.77	62.37/67.11	249.48/268.44
Pareek [Pareek *et al.*, 2013]	160	920	5650

Table 3. Encryption throughput and number of cycles for one encrypted byte.

Proposed Cryptosystem	ET in MBps	Number of Cycles Per Byte
Proposed V1	93.817/71.486	31.51/41.35
Proposed V2-8 bits	140.776/133.114	21/22.21
Proposed V2-32 bits	45.347/39.359	65.19/75.11
Zhang 1 [Zhang *et al.*, 2013]	25/25	122.07/122.07
Zhang 2 [Zhang *et al.*, 2013]	25/22.72	122.07/134.27
Wang [Wang *et al.*, 2011]	24.06/22.35	122.85/132.24
Akhshani [Akhshani *et al.*, 2012]	13.02	194.83
Wong [Wong *et al.*, 2008]	12.03/11.18	245.7/264.38
Pareek [Pareek *et al.*, 2013]	0.585	1630

analysis of the proposed cryptosystem is proved in the previous section in terms of a mathematical cryptanalysis, and it will be proved in the next sections in terms of statistical attacks.

6.2. *Plaintext sensitivity attacks*

A cryptosystem should be sensitive to one bit change in the plaintext. This requirement is most important to resist the known plaintext and the chosen-plaintext attacks [Lian *et al.*, 2005; Mao *et al.*, 2004]. In a chosen-plaintext attack, more than one plaintext (with one-bit changes between them) is selected to analyze the difference between their corresponding ciphertexts. The measurement tool to test the sensitivity of any cryptosystem to these attacks is carried out as: Select P_1 as the first plain image, change one bit in P_1 and name it as P_2 (i.e. P_1 and P_2 are exactly the same except for one bit, this bit is chosen to be located in the beginning, middle or the end of the first block, the plaintext results are calculated as an average of these three cases). Then both images (P_1 and P_2) are encrypted using the same secret key. This encryption produces two cipher images C_1 and C_2. Most researchers use two security parameters to measure the resistance of any chaos-based cryptosystem for plaintext sensitivity attacks. These parameters are: The Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI), they are given by the following equations respectively:

$$\mathrm{NPCR} = \frac{1}{L \times C \times P} \times \sum_{p=1}^{P} \sum_{i=1}^{L} \sum_{j=1}^{C} D(i,j,p) \times 100\% \tag{26}$$

where

$$D(i,j,p) = \begin{cases} 0, & \text{if } C_1(i,j,p) = C_2(i,j,p) \\ 1, & \text{if } C_1(i,j,p) \neq C_2(i,j,p) \end{cases} \tag{27}$$

$$\mathrm{UACI} = \frac{1}{L \times C \times P \times 255} \times \sum_{p=1}^{P} \sum_{i=1}^{L} \sum_{j=1}^{C} |C_1 - C_2| \times 100\%. \tag{28}$$

The optimal NPCR value is 99.61%, and the optimal UACI value is 33.46% [Wu *et al.*, 2011; Maleki *et al.*, 2008]. The above tests are also used to measure the resistance of a cryptosystem against the differential attacks introduced by Eli Biham and Adi Shamir [Biham & Shamir, 1991].

In our opinion, the previous tests are not sufficient to ensure that the proposed cryptosystem is resistant against plaintext sensitivity attacks. A new measurement test called the Avalanche effect is used. When the input is changed slightly (for example, flipping a single bit) and the output is changed significantly (e.g. half of the output bits are flipped), in this case the Avalanche effect is achieved. The block cipher quality is achieved when there is a small change in either the key or the plaintext, and there will be a drastic change in the ciphertext [Mar & Latt, 2008]. Therefore, this evaluation test is used to measure the resistance of any cryptosystem to plaintext and key sensitivity attacks. The scenario of this test is exactly identical to the previous one, but here the Hamming Distance is used to test if the cryptosystem has the Avalanche effect or not:

$$\mathrm{HD}(C_1, C_2) = \frac{1}{|Ib|} \sum_{K=1}^{|Ib|} (C_1(K) \oplus C_2(K)) \tag{29}$$

with $|Ib| = 8 \times L \times C \times P$.

The Hamming Distance (HD) in bits between the corresponding ciphered images C_1 and C_2 should be close to 50% (probability of bit changes, one bit difference in plain image will make every bit of the corresponding cipher image change with a probability of a half [Wang *et al.*, 2013]). Therefore, the plaintext sensitivity attack would become a useless attacking method. All versions of our proposed cryptosystem are tested using 3000 random secret keys. For each execution, we calculated the HD, NPCR, and UACI values between the two ciphered images C_1 and C_2. Finally, the average of the previous tested values are calculated. As a result, all versions of our proposed cryptosystem achieve the Avalanche effect from the first round ($r = 1$) and then, they overcome the plaintext sensitivity attacks. The plain images under test were Lena, Barb, and Boat images of the same size 512 × 512 gray scale images (the selected images are chosen like those in the literature). Moreover, Lena and Peppers color images of the same size were used on the same test. Table 4 presents all of these results, and it is clear that the HD value is very close to the optimal value of 50% for the three proposed cryptosystems. Also the UACI and NPCR values are close to optimal. These values indicate that the proposed cryptosystems are very sensitive to one bit

Table 4. HD, UACI and NPCR plain-text sensitivity tests.

Proposed Cryptosystem	Image Name	Image Size	HD	UACI	NPCR
Proposed V1	Barb	$512 \times 512 \times 1$	0.499630	33.438	99.532
Proposed V2-8 bit	Barb	$512 \times 512 \times 1$	0.499975	33.462	99.611
Proposed V2-32 bit	Barb	$512 \times 512 \times 1$	0.499978	33.460	99.609
Zhang 1 [Zhang *et al.*, 2013]	Barb	$512 \times 512 \times 1$	/	33.475	99.663
Zhang 2 [Zhang *et al.*, 2013]	Barb	$512 \times 512 \times 1$	/	33.420	99.582
Proposed V1	Lena	$512 \times 512 \times 1$	0.499587	33.437	99.521
Proposed V2-8 bit	Lena	$512 \times 512 \times 1$	0.499986	33.463	99.611
Proposed V2-32 bit	Lena	$512 \times 512 \times 1$	0.499975	33.459	99.609
Pareek [Pareek *et al.*, 2013]	Lena	$512 \times 512 \times 1$	/	31.79	99.6
Wong [Wong *et al.*, 2008]	Lena	$512 \times 512 \times 1$	/	32.82	99.44
Wang [Wang *et al.*, 2011]	Lena	$512 \times 512 \times 1$	/	33.435	99.607
Proposed V1	Boat	$256 \times 256 \times 1$	0.499576	33.434	99.524
Proposed V2-8 bit	Boat	$256 \times 256 \times 1$	0.499993	33.461	99.611
Proposed V2-32 bit	Boat	$256 \times 256 \times 1$	0.499955	33.466	99.609
Song [Song *et al.*, 2013]	Boat	$256 \times 256 \times 1$	33.453	99.625	/
Akhshani [Akhshani *et al.*, 2012]	Boat	$256 \times 256 \times 1$	0.499900	33.200	/
Proposed V1	Lena	$512 \times 512 \times 3$	0.499823	33.454	99.579
Proposed V2-8 bit	Lena	$512 \times 512 \times 3$	0.500001	33.466	99.611
Proposed V2-32 bit	Lena	$512 \times 512 \times 3$	0.499981	33.466	99.610
Proposed V1	Peppers	$512 \times 512 \times 3$	0.499853	33.451	99.576
Proposed V2-8 bit	Peppers	$512 \times 512 \times 3$	0.500035	33.466	99.610
Proposed V2-32 bit	Peppers	$512 \times 512 \times 3$	0.500001	33.463	99.609

change in the plaintext. Hence, a high security level is reached.

6.3. *Key sensitivity attack*

Key sensitivity is extremely crucial for any cryptosystem. A cryptosystem has a high security level relative to key sensitivity attacks if a slight change in the secret key will produce a completely different ciphered image [Pareek *et al.*, 2013]. The testing scenario of key sensitivity is almost identical to the plaintext sensitivity attack test: we have a plaintext P and two secret keys are different in one bit. First, P is encrypted using K_1 to obtain C_1. Then the same plaintext P is encrypted using K_2 to obtain C_2. Then the previous equations of the NPCR, UACI and HD, (26), (28) and (29) are used to evaluate the key sensitivity attacks of the proposed cryptosystem.

Table 5 presents the results obtained from the key sensitivity attack test for the three versions of the proposed cryptosystem using the same parameters as were used in Table 4. From Table 5, it is clear that the proposed cryptosystem has a high security level relative to the key sensitivity attacks.

6.4. *Histogram analysis*

One of the most common cryptosystem attacks is the one based on statistical analysis. A cryptosystem is considered to be strong against these attacks if the histogram of the encrypted image is uniformly distributed. In Fig. 8, we show some visual results obtained with the third version of the proposed cryptosystem (similar results are obtained for the other versions): (a) the plain Lena image of size $512 \times 512 \times 3$, (b) the corresponding cipher image, (c) the histogram of the plain image, and (d) its corresponding cipher image. The histogram of the encrypted image is very close to the uniform distribution and completely different from the plain image histogram. This means that there is no visual information than can be observed from the ciphered image of the proposed cryptosystem. The visual test is necessary but is not sufficient. To ensure uniformity, the chi-square test is applied [using Eq. (30)] to statistically confirm the uniformity of the histogram:

$$\chi^2_{\text{exp}} = \sum_{i=0}^{Nv-1} \frac{(o_i - e_i)^2}{e_i}. \tag{30}$$

Table 5. HD, UACI and NPCR key sensitivity tests.

Proposed Cryptosystem	Image Name	Image Size	HD	UACI	NPCR
Proposed V1	Barb	512 × 512 × 1	0.500029	33.464	99.610
Proposed V2-8 bit	Barb	512 × 512 × 1	0.499980	33.463	99.609
Proposed V2-32 bit	Barb	512 × 512 × 1	0.499989	33.461	99.609
Proposed V1	Lena	512 × 512 × 1	0.500014	33.463	99.610
Proposed V2-8 bit	Lena	512 × 512 × 1	0.499995	33.464	99.608
Proposed V2-32 bit	Lena	512 × 512 × 1	0.499952	33.465	99.608
Proposed V1	Boat	256 × 256 × 1	0.500015	33.462	99.609
Proposed V2-8 bit	Boat	256 × 256 × 1	0.499988	33.462	99.608
Proposed V2-32 bit	Boat	256 × 256 × 1	0.499995	33.460	99.609
Proposed V1	Lena	512 × 512 × 3	0.500007	33.464	99.610
Proposed V2-8 bit	Lena	512 × 512 × 3	0.499992	33.463	99.609
Proposed V2-32 bit	Lena	512 × 512 × 3	0.499994	33.465	99.609
Proposed V1	Peppers	512 × 512 × 3	0.500001	33.465	99.609
Proposed V2-8 bit	Peppers	512 × 512 × 3	0.499987	33.461	99.610
Proposed V2-32 bit	Peppers	512 × 512 × 3	0.499998	33.465	99.609

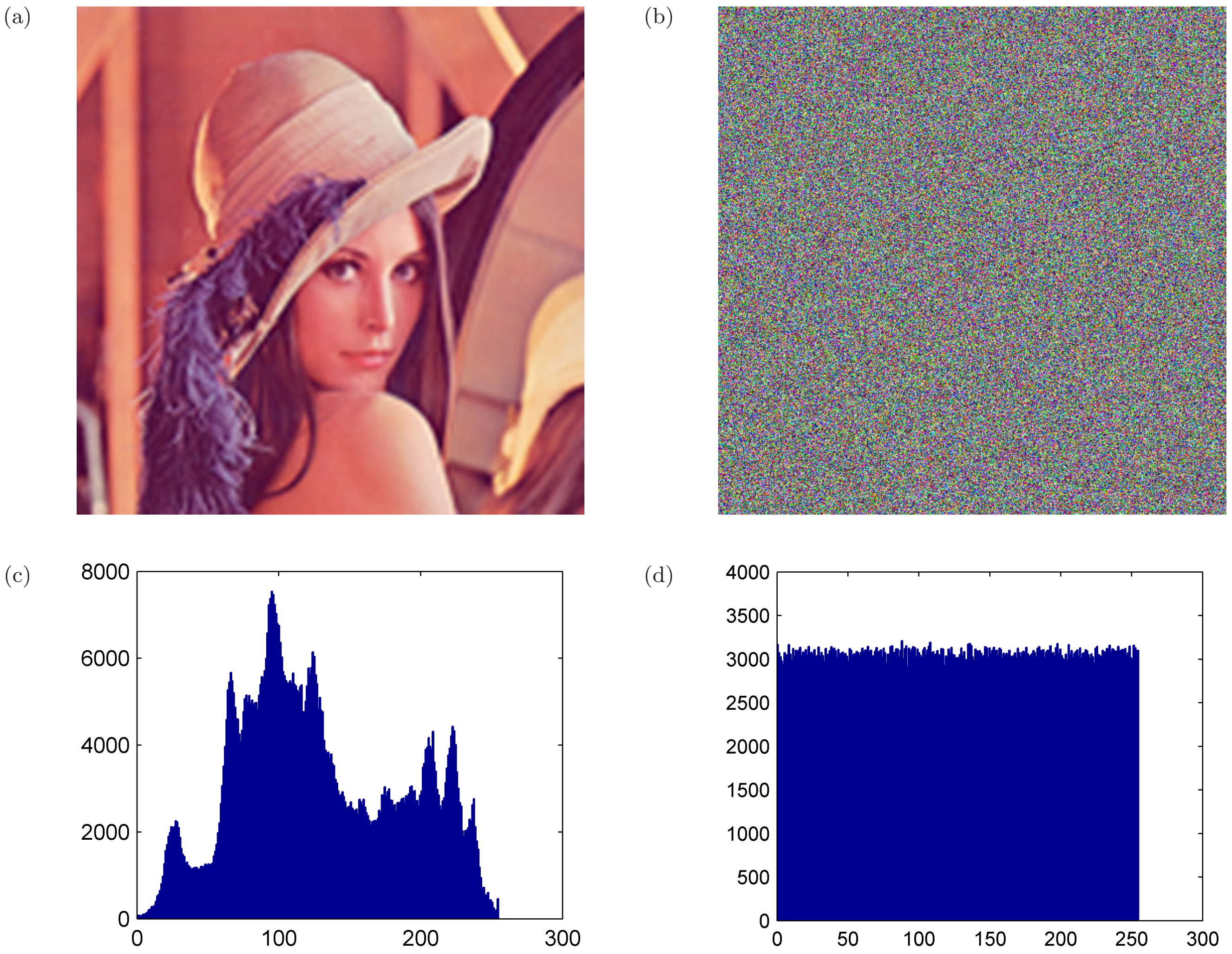


Fig. 8. Lena image and its ciphered version and their corresponding histograms. (a) Plain Lena image, (b) ciphered Lena image, (c) histogram of the plain Lena image and (d) histogram of the ciphered Lena image.

Table 6. Chi-square results.

Crypto Version	Ciphered Image	Chi-Square
Proposed V1	Lena	256.27
	Boat	262.46
	Baboon	259.91
Proposed V2-8 bit	Lena	259.63
	Boat	254.69
	Baboon	258.78
Proposed V2-32 bit	Lena	253.12
	Boat	252.44
	Baboon	253.14

Where Nv is the number of levels (here 256), o_i is the observed occurrence frequency of each color level (0–255) on the histogram of the ciphered image, and e_i is the expected occurrence frequency of the uniform distribution, given here by $e_i = \frac{L\times C\times P}{256}$. We present in Table 6 the results obtained from the chi-square test of histograms for three ciphered images of different nature (i.e. Lena, Boat, and Baboon). All of them have the same size of 128 × 128 × 3, with a significant level of 0.05. From the obtained values, we can observe that $\chi^2_{\text{exp}} < \chi^2_{\text{th}}(255, 0.05) = 293$, and then the tested histograms are uniform and do not reveal any information for statistical analysis.

6.5. *Correlation analysis*

Correlation analysis is also one of the types of statistical attacks. Correlation analysis should not give any information on the secret key used or any partial information of the original plain image. This means that the encrypted image should be greatly different from its original one. Correlation analysis is one of the usual ways to measure this property.

Table 7. Correlation analysis results.

Cryptosystem Name	Horizontal	Vertical	Diagonal
Proposed Algorithm V1	0.0085	0.0097	0.0092
Proposed Algorithm V2-8 bit	0.0087	0.0098	0.0096
Proposed Algorithm V2-32 bit	0.0087	0.0098	0.0096

Indeed, it is well known that adjacent pixels in the plain images are highly redundant and correlated. So, in the encrypted images, adjacent pixels should have a redundancy and a correlation as low as possible. To test the correlation between adjacent pixels, the following procedure was carried out. Firstly, 8000 pairs of two adjacent pixels are selected randomly in vertical, horizontal, and diagonal directions from the original and the encrypted images. Then, the correlation coefficient is computed according to the following formulas:

$$\rho_{xy} = \frac{\text{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{31}$$

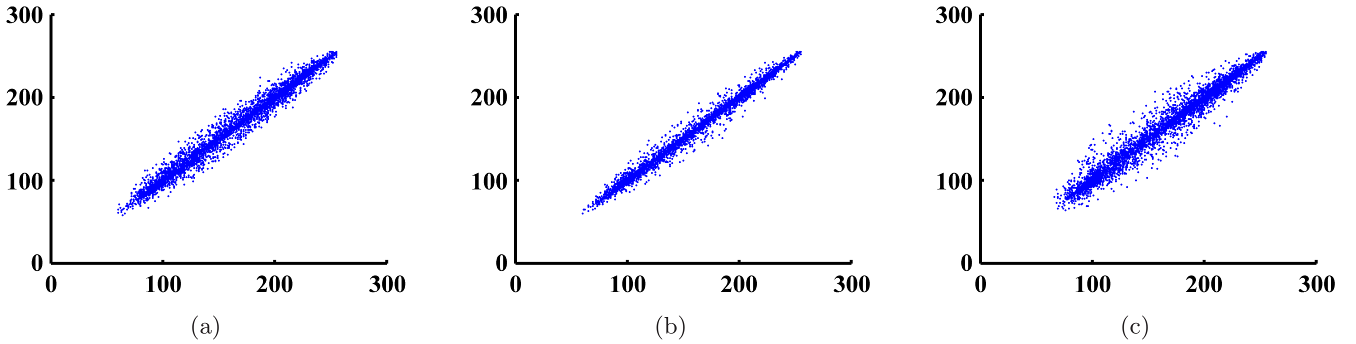
where

$$\text{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}([x_i - E(x)][y_i - E(y)]) \tag{32}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{33}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i) \tag{34}$$

$N = 8000$ is the sample size, while x and y are the gray-level values of the two adjacent pixels in the



(a)

(b)

(c)

Fig. 9. Correlation analysis of Lena and its ciphered image in three directions. (a) Horizontal correlation of the plain image, (b) vertical correlation of the plain image, (c) diagonal correlation of the plain image, (d) horizontal correlation of the ciphered image, (e) vertical correlation of the ciphered image and (f) diagonal correlation of the ciphered image.

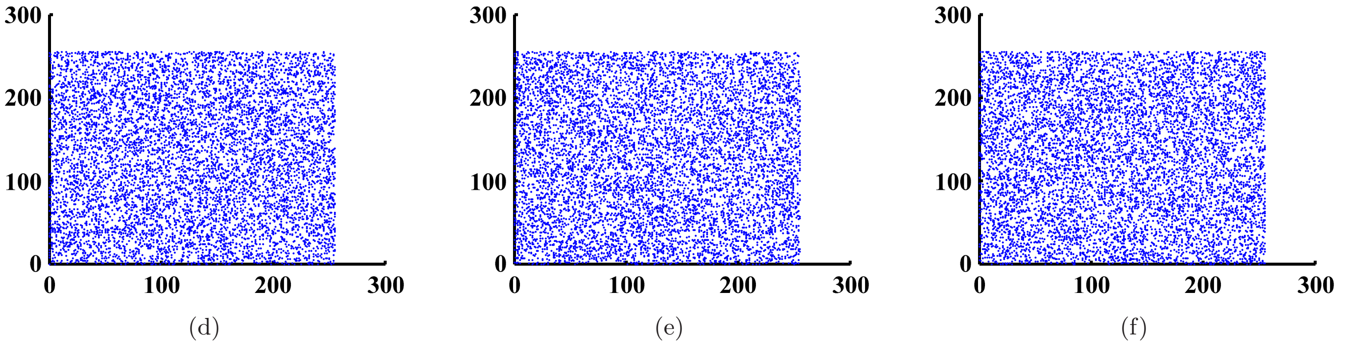


Fig. 9. (*Continued*)

image. One example of this extensive study is the Barb gray scale image of 512×512×1. The obtained results are shown in Table 7 and Fig. 9. These results demonstrate that the correlation coefficient, in all directions, of the plain images is close to one (see Fig. 9), and the correlation coefficient of the encrypted images is close to zero. This means that there is no detectable correlation between the original and its corresponding ciphered image, and also, there is no relation between pixels of the ciphered image.

7. Conclusion

In this paper, firstly we studied and analyzed one of the fastest chaos-based cryptosystems, namely Zhang cryptosystems. Then, based on a similar structure of the Zhang and Fridrich cryptosystems, we designed three versions of a chaos-based cryptosystem. We have shown that all versions of our proposed cryptosystem are faster and more secure than Zhang and many other chaos-based cryptosystems. The time performance is carried out using encryption/decryption time, running speed, and the number of cycles needed to encrypt or decrypt one byte. The last measurement method is necessary to compare different cryptosystems working on different platforms. All versions of our proposed cryptosystem are implemented using the CBC mode and a robust chaotic generator to produce the dynamic keys for each new encryption round and new block. The high security level of all versions of the proposed cryptosystem is verified by testing them for different kinds of known mathematical attacks, and using the well-known statistical analysis. Finally, all results prove the superiority of the proposed cryptosystems for use in secure and real-time applications.

Acknowledgment

This work is supported by the European Celtic-Plus project 4KREPROSYS — 4K ultraHD TV wireless REmote PROduction SYStems.

References

Abd El-Latif, A. A., Niu, X. & Amin, M. [2012] “A new image cipher in time and frequency domains,” *Opt. Commun.* **285**, 4241–4251.

Akhshani, A., Akhavan, A., Lim, S.-C. & Hassan, Z. [2012] “An image encryption scheme based on quantum logistic map,” *Commun. Nonlin. Sci. Numer. Simul.* **17**, 4653–4661.

Behnia, S., Akhshani, A., Mahmodi, H. & Akhavan, A. [2008] “A novel algorithm for image encryption based on mixture of chaotic maps,” *Chaos Solit. Fract.* **35**, 408–419.

Bhargava, B., Shi, C. & Wang, S.-Y. [2004] “MPEG video encryption algorithms,” *Multimed. Tools Appl.* **24**, 57–79.

Bhatnagar, G. & Jonathan Wu, Q. [2012] “Selective image encryption based on pixels of interest and singular value decomposition,” *Dig. Sign. Process.* **22**, 648–663.

Biham, E. & Shamir, A. [1991] “Differential cryptanalysis of DES-like cryptosystems,” *J. Cryptology* **4**, 3–72.

Chang, W.-D. [2009] “Digital secure communication via chaotic systems,” *Dig. Sign. Process.* **19**, 693–699.

Chen, G., Mao, Y. & Chui, C. K. [2004] “A symmetric image encryption scheme based on 3D chaotic cat maps,” *Chaos Solit. Fract.* **21**, 749–761.

Chen, J.-X., Zhu, Z.-L., Fu, C., Yu, H. & Zhang, L.-B. [2015] “A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism,” *Commun. Nonlin. Sci. Numer. Simul.* **20**, 846–860.

Chiaraluce, F., Ciccarelli, L., Gambi, E., Pierleoni, P. & Reginelli, M. [2002] “A new chaotic algorithm for video encryption,” *IEEE Trans. Consum. Electron.* **48**, 838–844.

Ehrsam, W. F., Meyer, C. H., Smith, J. L. & Tuchman, W. L. [1978] "Message verification and transmission error detection by block chaining," US Patent 4,074,066.

El Assad, S. & Noura, H. [2011] "Generator of chaotic sequences and corresponding generating system," US Patent App. 13/638, 126.

El Assad, S., Farajallah, M. & Vladeanu, C. [2014] "Chaos-based block ciphers: An overview," *10th IEEE Int. Conf. Communications* (*COMM*) (IEEE), pp. 1–4.

Farajallah, M., El Assad, S. & Chetto, M. [2013a] "Dynamic adjustment of the chaos-based security in real-time energy harvesting sensors," *IEEE Int. Conf. Green Computing and Communications* (*GreenCom*), *Internet of Things* (*iThings/CPSCom*), *Cyber*, *Physical and Social Computing*, pp. 282–289.

Farajallah, M., Fawaz, Z., El Assad, S. & Deforges, O. [2013b] "Efficient image encryption and authentication scheme based on chaotic sequences," *7th IEEE Int. Conf. Emerging Security Information*, *Systems and Technologies* (*SECURWARE*), pp. 150–155.

Fridrich, J. [1997] "Image encryption based on chaotic maps," *IEEE Int. Conf. Systems*, *Man*, *and Cybernetics*, *Computational Cybernetics and Simulation*, pp. 1105–1110.

Fridrich, J. [1998] "Symmetric ciphers based on two-dimensional chaotic maps," *Int. J. Bifurcation and Chaos* **8**, 1259–1284.

Furht, B. & Socek, D. [2003] "Multimedia security: Encryption techniques," *IEC Comprehensive Report on Information Security*, pp. 335–349.

Hamidouche, W., Farajallah, M., Raulet, M., Deforges, O. & El Assad, S. [2015] "Selective video encryption using chaotic system in the SHVC extension," *40th IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*), pp. 1762–1766.

Kassem, A., Al Haj Hassan, H., Harkouss, Y. & Assaf, R. [2014] "Efficient neural chaotic generator for image encryption," *Dig. Sign. Process.* **25**, 266–274.

Li, S., Chen, G. & Zheng, X. [2006] *Chaos-Based Encryption for Digital Image and Video* (CRC Press).

Lian, S., Sun, J. & Wang, Z. [2005] "Security analysis of a chaos-based image encryption algorithm," *Physica A* **351**, 645–661.

Lian, S., Sun, J., Wang, J. & Wang, Z. [2007] "A chaotic stream cipher and the usage in video protection," *Chaos Solit. Fract.* **34**, 851–859.

Maleki, F., Mohades, A., Hashemi, S. M. & Shiri, M. E. [2008] "An image encryption system by cellular automata with memory," *3rd IEEE Int. Conf. Availability*, *Reliability and Security* (*ARES*), pp. 1266–1271.

Mansour, I., Chalhoub, G. & Bakhache, B. [2012] "Evaluation of a fast symmetric cryptographic algorithm based on the chaos theory for wireless sensor networks," *11th IEEE Int. Conf. Trust*, *Security and Privacy in Computing and Communications* (*TrustCom*), pp. 913–919.

Mao, Y., Chen, G. & Lian, S. [2004] "A novel fast image encryption scheme based on 3D chaotic Baker maps," *Int. J. Bifurcation and Chaos* **14**, 3613–3624.

Mar, P. P. & Latt, K. M. [2008] "New analysis methods on strict Avalanche criterion of S-boxes," *World Acad. Sci. Engin. Technol.* **48**, 150–154.

Masuda, N. & Aihara, K. [2002] "Cryptosystems with discretized chaotic maps," *IEEE Trans. Circuits Syst.-I*: *Fund. Th. Appl.* **49**, 28–40.

Masuda, N., Jakimoski, G., Aihara, K. & Kocarev, L. [2006] "Chaotic block ciphers: From theory to practical algorithms," *IEEE Trans. Circuits Syst.-I*: *Reg. Papers* **53**, 1341–1352.

Murillo-Escobar, M., Cruz-Hernández, C., Abundiz-Pérez, F., López-Gutiérrez, R. & Del Campo, O. A. [2015] "A RGB image encryption algorithm based on total plain image characteristics and chaos," *Sign. Process.* **109**, 119–131.

Pareek, N. K., Patidar, V. & Sud, K. K. [2013] "Diffusion–substitution based gray image encryption scheme," *Dig. Sign. Process.* **23**, 894–901.

Rukhin, A., Soto, J., Nechvatal, J., Smid, M. & Barker, E. [2001] "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Tech. Rep., DTIC Document.

Shannon, C. E. [1949] "Communication theory of secrecy systems," *Bell Syst. Techn. J.* **28**, 656–715.

Socek, D., Li, S., Magliveras, S. S. & Furht, B. [2005] "Enhanced 1-D chaotic key based algorithm for image encryption," *1st IEEE Int. Conf. Security and Privacy for Emerging Areas in Communications Networks* (*SecureComm*), pp. 406–407.

Solak, E., Çokal, C., Yildiz, O. T. & Biyikoğlu, T. [2010] "Cryptanalysis of Fridrich's chaotic image encryption," *Int. J. Bifurcation and Chaos* **20**, 1405–1413.

Song, C.-Y., Qiao, Y.-L. & Zhang, X.-Z. [2013] "An image encryption scheme based on new spatiotemporal chaos," *Optik — Int. J. Light and Electron Opt.* **124**, 3329–3334.

Wang, Y., Wong, K.-W., Liao, X. & Chen, G. [2011] "A new chaos-based fast image encryption algorithm," *Appl. Soft Comput.* **11**, 514–522.

Wang, X., Luan, D. & Bao, X. [2013] "Cryptanalysis of an image encryption algorithm using Chebyshev generator," *Dig. Sign. Process.* **25**, 244–247.

Wong, K.-W., Kwok, B. S.-H. & Law, W.-S. [2008] "A fast image encryption scheme based on chaotic standard map," *Phys. Lett. A* **372**, 2645–2652.

Wu, Y., Noonan, J. P. & Agaian, S. [2011] "NPCR and UACI randomness tests for image encryption," *Cyber J.: Multidisciplinary J. Sci. Technol., J. Selected Areas in Telecommun.*, 31–38.

Yang, H., Wong, K.-W., Liao, X., Zhang, W. & Wei, P. [2010] "A fast image encryption and authentication scheme based on chaotic maps," *Commun. Nonlin. Sci. Numer. Simul.* **15**, 3507–3517.

Zhang, L., Liao, X. & Wang, X. [2005] "An image encryption approach based on chaotic maps," *Chaos Solit. Fract.* **24**, 759–765.

Zhang, W., Wong, K.-W., Yu, H. & Zhu, Z.-L. [2013] "An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion," *Commun. Nonlin. Sci. Numer. Simul.* **18**, 2066–2080.