Hindawi
Security and Communication Networks
Volume 2018, Article ID 1840207, 20 pages
https://doi.org/10.1155/2018/1840207

Research Article

A Robust Watermarking Scheme for Online Multimedia Copyright Protection Using New Chaotic Map

Amir Anees,[1] **Iqtadar Hussain**,[2] **Abdulmohsen Algarni**,[3] **and Muhammad Aslam**[4]

[1]*Department of Electrical Engineering, HITEC University, Pakistan*
[2]*Department of Mathematics, College of Science, King Khalid University, Abha, Saudi Arabia*
[3]*College of Computer Science, King Khalid University, Abha, Saudi Arabia*
[4]*Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan*

Correspondence should be addressed to Amir Anees; a.anees@latrobe.edu.au

Received 22 November 2017; Accepted 14 May 2018; Published 28 June 2018

Academic Editor: Salvatore D'Antonio

The protection of copyrights of digital media uploaded to the Internet is a growing problem. In this paper, first, we present a unified framework for embedding and detecting watermark in digital data. Second, a new robust watermarking scheme is proposed considering this concern. The proposed work incorporates three chaotic maps which specify the location for embedding the watermark. Third, a new chaotic map, the Extended Logistic map, is proposed in this work. The proposed map has a bigger range than logistic and cubic maps. It has shown good results in a bifurcation, sensitivity to initial conditions, and randomness tests. Furthermore, with the detailed analysis of initial parameters, it is justified that Extended Logistic map can be used in secure communication, particularly watermarking. Fourth, to check the robustness of proposed watermarking scheme, we have done a series of analyses and standard attacks. The results confirm that the proposed watermarking scheme is robust against visual and statistical analysis and can resist the standard attacks.

1. Introduction

With the exponential growth of multimedia and their applications, most of the digital data is exchanged thorough the insecure channel [1]. It is important to protect the digital data, in particular secret data. To ensure the security of digital data, much effort has been put in the area of secure communication [2–4]. At the top algorithmic level, secure communication can be divided into three categories: cryptography [5–7], steganography, and watermarking [8–10]. The purpose of cryptography and stenography is to conceal the secret data; this is why these fields also lie in the information hiding subcategory [11]. Although the goal is the same, the methods are differently employed in cryptography and steganography. In cryptography, the data is transformed into an unreadable format so that, after transformation (encryption), the attacker can not deduce any information from it [12]. In steganography, the data is inserted or embedded into a carrier which can be a text, audio, image, or video. It is expected that the texture of carrier before and after the insertion of data will remain the same so that the attacker can not deduce any information regarding the data from the carrier [13, 14]. The methodology of steganography and watermarking is same, but the purposes are different. Steganography deals with the data hiding and watermarking concerns with the copyright protection of data [15, 16]. Instead of secret data, a watermark is inserted into the carrier to claim the copyrights when needed. Let us suppose that C denotes the carrier and w represents the watermark to be inserted and let Δ be the watermarking function, then the watermarked W data is given as

$$W = \Delta(C, w). \tag{1}$$

At the time of claiming the copyrights, inverse watermarking function Δ^{-1} is applied on W to extract the watermark as follows:

$$w = \Delta^{-1}(W). \tag{2}$$

This kind of watermarking scheme is known as private watermarking in which only W is required to extract the

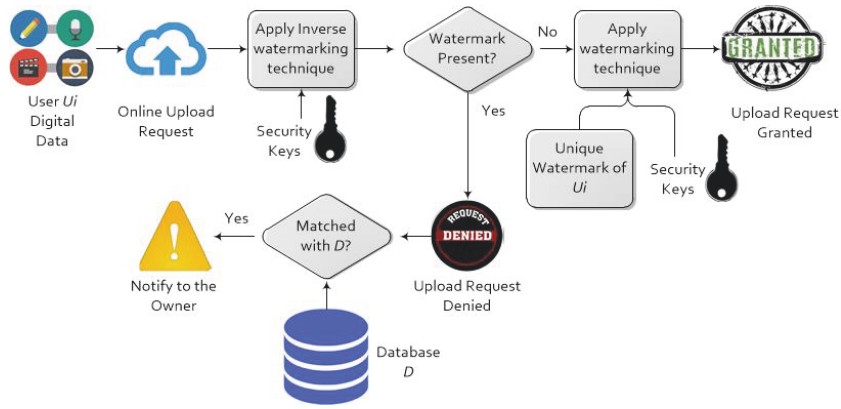


Figure 1: A top level block description of the proposed unified framework for protecting the rightful ownership of digital data. If an unauthenticated user tries to download and upload a data of user U_i, not only will the permission be denied but the domain will also notify U_i to legally make a case against that unauthenticated user.

watermark. However, some watermarking schemes do need the C as well with the W as defined as follows:

$$w = \Delta^{-1}(W, C). \tag{3}$$

This sort of watermarking scheme is known as public watermarking. In this manuscript, we have employed the private watermarking scheme using three chaotic maps which serve the functionality of Δ.

1.1. Motivations and Proposed Framework. Although there have been tremendous works on watermarking for the copyrights protection still the practical and real-life applications do need much attention, specifically in the area of online privacy of digital data. For example, a public domain of video uploader, youtube.com, does not have the sophisticated framework for the protection of rightful ownership. Let say, for instance, Bob has uploaded his video and after some time (days), Alice has downloaded that video and uploaded it again by her name. At the time of uploading, Alice has been granted the permission of uploading that video which should not be the case. The management of that public domain eventually remove that video after some days, and if the name of the video uploaded by Alice is significantly different from the one given by Bob, then it can take even more time to look up for that video and eventually in removing the video. Similarly, other than videos, piracy of digital images face the same problem.

In this work, we have proposed a unified framework for protecting the rightful ownership of digital data. The proposed framework is shown in Figure 1. It works as follows: let say a user U_i wants to upload his data on a public domain. First, he requests that domain upload that data. The public domain upon receiving the request applies the proposed inverse watermarking function Δ^{-1} to see if there is any watermark present in it. If there is any watermark present in that data, the domain will deny the upload request. Furthermore, the public domain will match the extracted watermark with the database. If the extracted watermark is matched with someone else watermark, the domain will notify the rightful owner of that data so that he or she can move with the legal case. In the case in which there is no watermark present in it, the domain will grant the permission of uploading that data and also will apply the proposed watermarking function Δ with the unique watermark w_i associated with the user U_i. In that way, let say, in future, if Alice tries to download and upload that data, not only will the permission be denied, but the domain will also notify U_i to make a case against her legally.

1.2. Contributions of This Work. In this paper, we develop a framework for watermarking using three different chaotic maps. The proposed work is effective and efficient as evident from the different analysis and attacks examined. The contributions of this paper are summarized as follows:

(1) We have proposed a unified framework for the protection of rightful ownership. In this framework, the wrong owner is not only denied but also reported to the right owner for any legal case.

(2) We have proposed a new chaotic map, Extended Logistic map in this work. The proposed map has a bigger range than logistic and cubic maps. It has shown good results in a bifurcation, sensitivity to initial conditions, and randomness tests. Furthermore, with the detailed analysis of initial parameters, it is justified that Extended Logistic map can be used in secure communication, particularly watermarking.

(3) We have proposed a watermarking scheme for any digital data specifically image data. The proposed watermarking is primarily based on the embedding of a watermark in lease significant bits of the carrier using three different chaotic maps. The initial conditions of these three chaotic maps are considered as secret keys [19–22].

(4) We have done noise resistant analysis in which we have shown that if the watermarked image is corrupted by the channel noise or by an unauthorized user, the inverse watermarking algorithm can

correctly extract the watermark from the corrupted watermarked image with some minor changes.

(5) To evaluate the strength of proposed watermarking algorithm, we have done a series of analyses along with standard attacks. The results of these analyses showed the superior performance and robustness of our proposed watermarking algorithm.

The remaining part of the manuscript is planned as follows: Section 2 presents the new chaotic map with its significance, Section 3 is devoted to the proposed watermarking scheme in detail. Section 4 is dedicated to simulation results and statistical analysis; Section 5 presents the performance analysis that will evaluate the proposed scheme with standard analysis and attacks and Section 6 concludes the manuscript.

2. A New Chaotic Map and Its Significance in Watermarking

This section will briefly introduce a new chaotic map which is a combination of logistic and cubic chaotic maps. The importance of usage of the proposed chaotic map is illustrated through the bifurcation diagrams, randomness, and sensitivity tests. The basics of other chaotic maps employed in the proposed watermarking algorithm are also given. These maps are piecewise linear chaotic map and Tangent Delay Ellipse Reflecting Cavity Map System (TD-ERCS).

The logistic chaotic map is given as [23]

$$x_n = r x_{n-1} \left(1 - x_{n-1}\right). \tag{4}$$

where $x_0 \in (0, 1)$ and $r \in (0, 4)$ are the initial seed parameters.

The cubic chaotic map is given as [24]

$$x_n = r x_{n-1} \left(1 - x_{n-1}^2\right). \tag{5}$$

where $x_0 \in (0, 1)$ and $r \in (0, 2.6)$ are the initial seed parameters.

Combining the above two maps, the new proposed Extended Logistic (EL) chaotic map is given as

$$x_n = r x_{n-1} \left(1 - x_{n-1}\right)\left(c + x_{n-1}\right). \tag{6}$$

where $x_0 \in (0, 1)$, $c \in (0, \infty)$, and r which is c dependable are the initial seed parameters.

We have analyzed the bifurcation diagrams of the EL map considering different values of parameter c plotted in Figure 2. These figures are the combination of all the x-vectors plotted against r. The r values are plotted on x-axis with spacing of 0.01 and x values are plotted on y-axis for each and every r value. Furthermore, the x values are plotted after 500 iterations to see the long-term effect of EL map. It can be observed that, by increasing the parameter c, the chaotic region of EL map is shrinking. Nonetheless, with parameter c in addition to x and r, the key space is huge enough to resist brute force attack and thus is sufficient, relevant, and appropriate to be used in secure communication.

In addition, we have examined a bifurcation diagram shown in Figure 2(a) to observe the overall behavior of EL map. Based on Figure 2(a), the interval considering r can be divided into following segments:

(i) When $r \in (0, 1.2)$ for $c = 2$, the iteration sequence of EL map shows a stable behavior, that is, after few iterations, the sequence comes to a constant point. The stable behavior can be shown in Figure 3. Figure 3(a) shows the bifurcation diagram of EL map for $r = 0$ to $r = 1.2$ when $c = 2$; Figures 3(b)–3(f) demonstrate the iteration sequence of this chaotic map for initial conditions of $r = 0.2$ to $r = 1.0$, $x_0 = 0.5$, and $c = 2$ with a spacing of $r = 0.2$. It can be observed that the iteration sequences come to a constant point after some iterations and thus cannot be used in secure communication.

(ii) When $r \in (1.2, 1.41)$ for $c = 2$, the iteration sequence of EL map shows a periodic behavior; that is, after few iterations, the sequence iterates between two or four points. The periodic behavior can be shown in Figure 4. Figure 4(a) shows the bifurcation diagram of EL map for $r = 1.2$ to $r = 1.41$ when $c = 2$; Figures 4(b)–4(f) demonstrate the iteration sequence of this chaotic map for initial conditions of $r = 1.22$ to $r = 1.40$, $x_0 = 0.5$, and $c = 2$ with an approximate spacing of $r = 0.04$. It can be observed that the iteration sequences iterate between two or four points after some iterations and thus cannot be used in secure communication.

(iii) When $r \in (1.41, 1.60)$, the iteration sequence for EL map shows random points exhibiting a chaotic behavior. Figure 5(a) shows the bifurcation diagram of EL map for $r = 1.41$ to $r = 1.60$; Figures 5(b)–5(f) demonstrate the iteration sequence of this chaotic map for initial conditions of $r = 1.41$ to $r = 1.60$, $x_0 = 0.5$, and $c = 2$ with an approximate spacing of $r = 0.03$. As can be seen from the Figure 5(a), the whole interval does not produces chaotic behavior as further elaborated in Figure 5(d). Figure 5(d) demonstrates the iteration sequence of this chaotic map for initial conditions of $r = 1.51$, $x_0 = 0.5$, and $c = 2$. It can be observed that the iteration sequences iterate between two points after some iterations and thus cannot be used in secure communication. Nonetheless, majority of this interval does exhibit as chaotic behavior, the iteration sequences are completely random and thus can be used in secure communication.

The chaotic behavior is further tested using the standard NIST-800-22 statistical tests [25]. This test suite includes 15 statistical tests which were originally designed for a larger stream of binary bits (usually > 10000); however these tests can also be applied on shorter streams as well as on integer values. The results of these tests on EL map for initial conditions of $r = 1.48$, $x_0 = 0.5$, and $c = 2$ are listed in Table 1 showing that the generated chaotic sequences passes the randomness tests.

For the appropriate usage in secure communication, randomness alone is not enough for a chaotic map. One of the essential requirements for any random generator in security is to be sensitive to the initial conditions. We have plotted chaotic sequence for EL map for initial conditions of $r = 1.56$, $x_0 = 0.500000000$, and $c = 2$, shown in Figure 6(a); in the

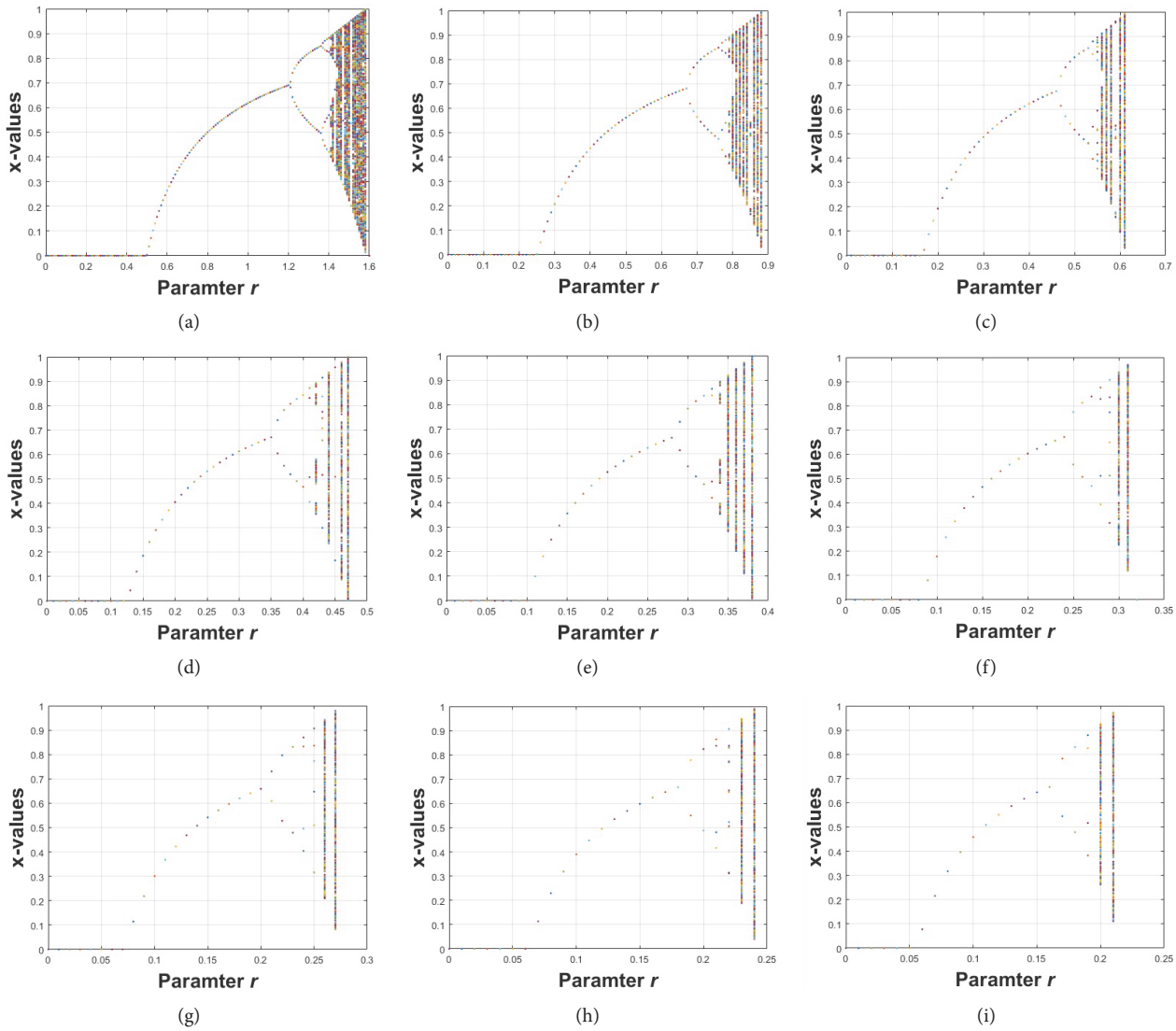


FIGURE 2: Bifurcation diagrams of the EL map considering different values of parameter c. (a) $c = 2$, (b) $c = 4$, (c) $c = 6$, (d) $c = 8$, (e) $c = 10$, (f) $c = 12$, (g) $c = 14$, (h) $c = 16$, and (i) $c = 18$. These figures are the combination of all the x-vectors plotted against r. The r values are plotted on x-axis with spacing of 0.01 and x values are plotted on y-axis for each and every r values. Furthermore, the x values are plotted after 500 iterations to see the long-term effect of EL map.

same figure, we have also plotted the chaotic sequences for EL map for slightly change initial conditions of $r = 1.56$, $x_0 = 0.500000001$, and $c = 2$ and $r = 1.56$, $x_0 = 0.500000002$, and $c = 2$. For the first 35 iterations, all the three sequences are same; however after that point, all sequences are completely out of phase to each other. Also, the EL map is sensitive to the initial conditions for r as well. Figure 6(b) shows the chaotic sequence for EL map for initial conditions of $r = 1.5600000000$, $x_0 = 0.5$, and $c = 2$ and in the same figure, two graphs are plotted for EL map for initial conditions of $r = 1.5600000001$, $x_0 = 0.5$, and $c = 2$ and $r = 1.5600000002$, $x_0 = 0.5$, and $c = 2$. It can be observed that at the beginning all the three sequences are almost same but with the increase in iterations; all the sequences can be recognized individually.

It can be shown that the other two chaotic maps, TD-ERCS and piecewise linear map, also have the same properties as EL map. The TD-ERCS map gives two chaotic sequences, x and k; mathematically, it is given as [26]

$$x_n = -\frac{2k_{n-1}y_{n-1} + x_{n-1}\left(\mu^2 - k_{n-1}^2\right)}{\mu^2 + k_{n-1}^2}.$$
$$k_n = -\frac{2k'_{n-m} - k_{n-1} + k_{n-1}k'^2_{n-m}}{1 + 2k_{n-1}k'_{n-m} - k'^2_{n-m}} \tag{7}$$

where

$$k'_n = -\frac{x_n}{x_n}\mu^2.$$

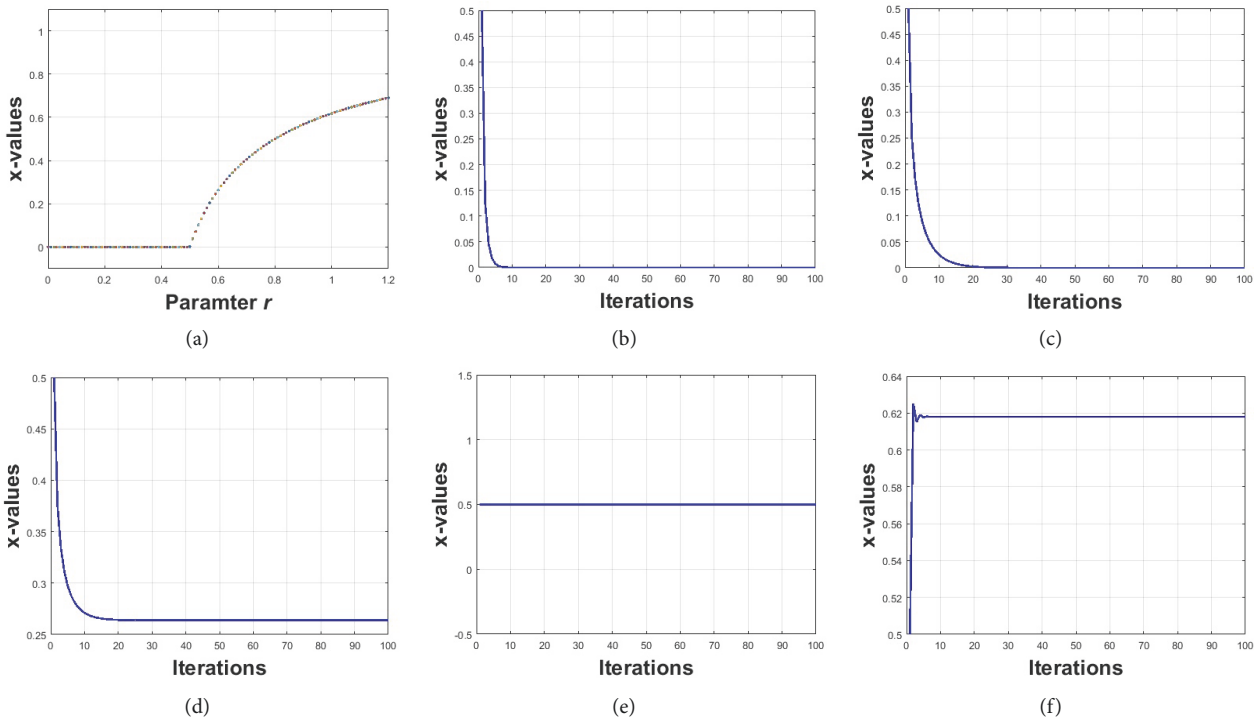


FIGURE 3: (a) Bifurcation diagram of EL map for $r = 0.2$ to $r = 1.0$, with initial conditions of $x_0 = 0.5$ and $c = 2$; (b)-(f) the iteration sequence diagrams of EL map for initial conditions of $r = 0.2$ to $r = 1.0$, $x_0 = 0.5$, and $c = 2$ with a spacing of $r = 0.2$. It can be observed that the iteration sequences come to a constant point after some iterations and thus cannot be used in secure communication.

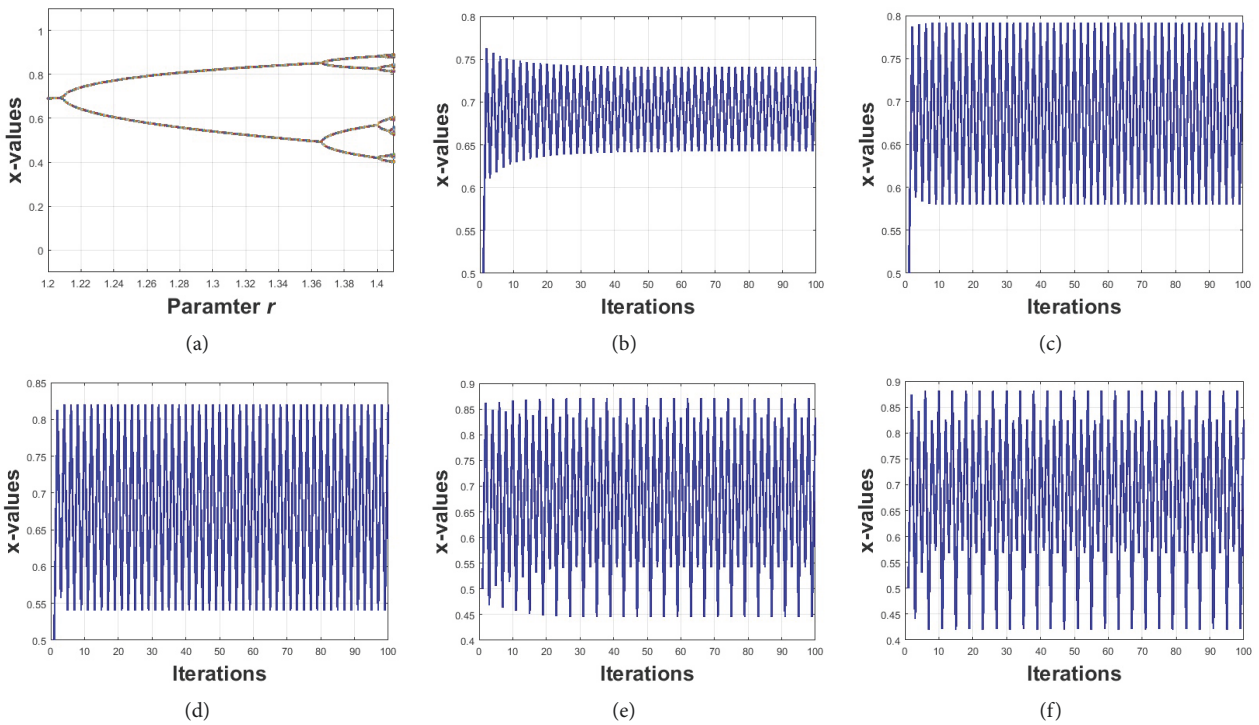


FIGURE 4: (a) Bifurcation diagram of EL map for $r = 1.2$ to $r = 1.41$, with initial conditions of $x_0 = 0.5$ and $c = 2$; (b)-(f) the iteration sequence diagrams of EL map for initial conditions of $r = 1.22$ to $r = 1.40$, $x_0 = 0.5$, and $c = 2$ with an approximate spacing of $r = 0.04$. It can be observed that the iteration sequences iterate between two or four points after some iterations and thus cannot be used in secure communication.

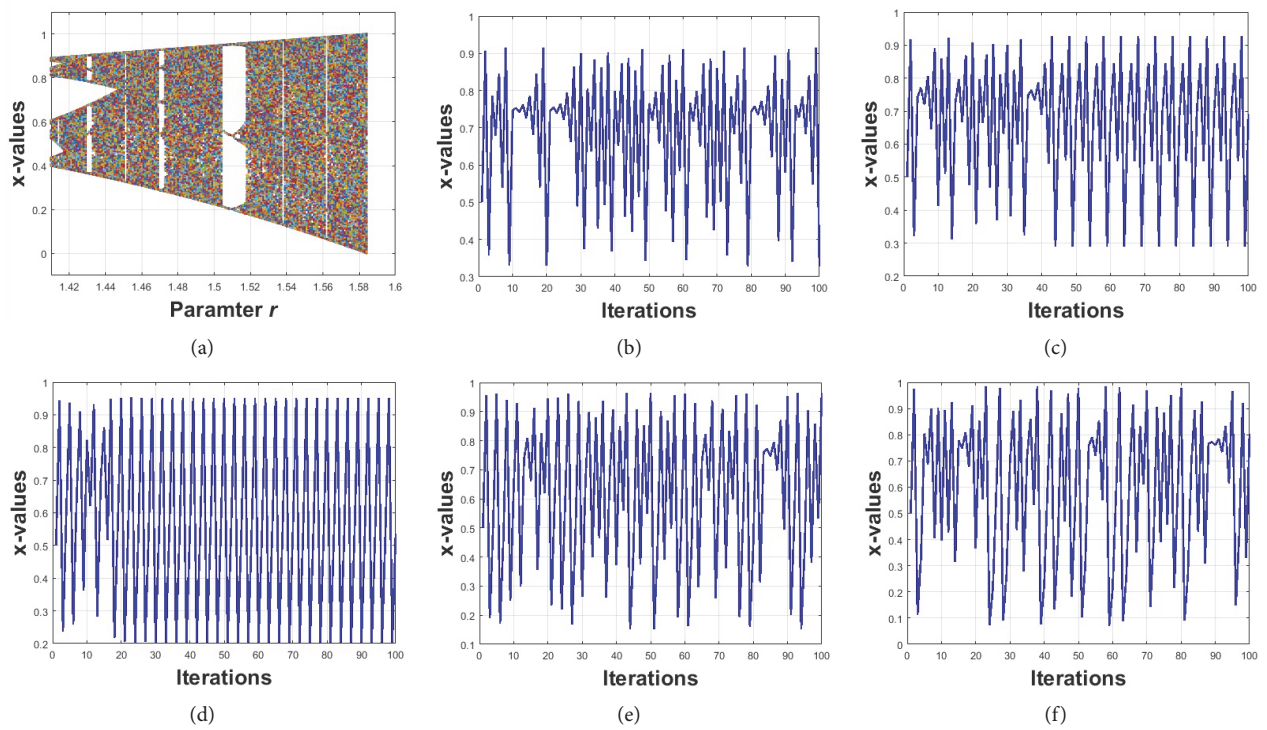
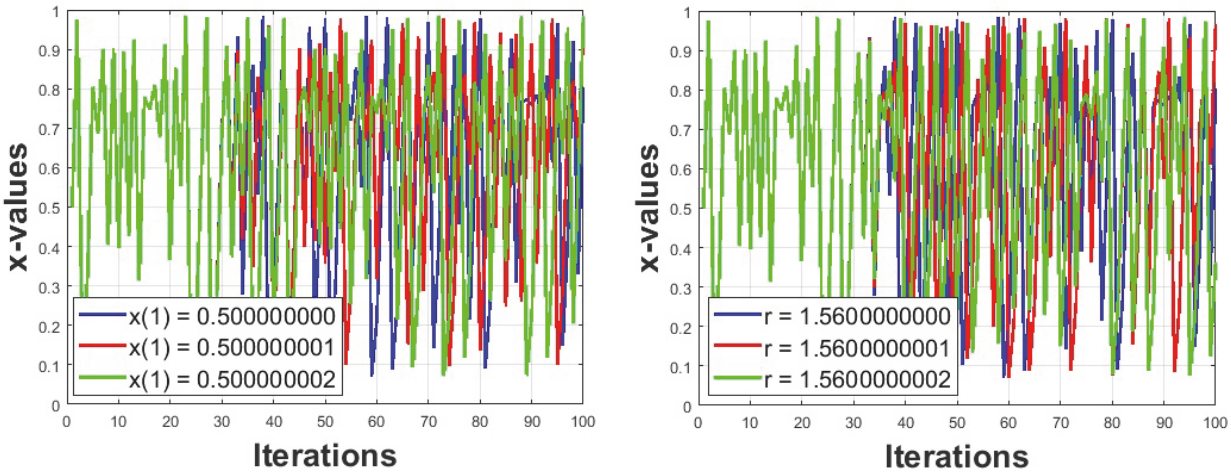


FIGURE 5: (a) Bifurcation diagram of EL map for $r = 1.41$ to $r = 1.60$, with initial conditions of $x_0 = 0.5$ and $c = 2$; (b)-(f) the iteration sequence diagrams of EL map for initial conditions of $r = 1.41$ to $r = 1.60$, $x_0 = 0.5$, and $c = 2$ with an approximate spacing of $r = 0.03$. The whole interval does not produce chaotic behavior as elaborated in (d). Nonetheless, majority of this interval does exhibit as chaotic behavior; the iteration sequences are completely random and thus can be used in secure communication.

TABLE 1: NIST statistical tests to check the randomness of chaotic sequences generated by EL map for initial conditions of $r = 1.48$, $x_0 = 0.5$, and $c = 2$. The results show that the chaotic sequences passes the randomness statistical tests.

Statistical Test	*P* Value	Decision
Frequency (Mono Bit) Test	0.9984	Passed
Frequency Test within a Block		
(Blocks sizes: 3, 4, 5, 6, 7, 8)	0.6845	Passed
The Runs Test	0.0409	Passed
Tests for the Longest-Run-of-Ones in a Block		
(Blocks size: 8)	0.3089	Passed
The Binary Matrix Rank Test		
(4 Matrices, Rows: 8, Columns: 8)	0.0886	Passed
(16 Matrices, Rows: 4, Columns: 4)	0.1547	Passed
The Discrete Fourier Transform (Spectral) Test	0.1007	Passed
The Non-overlapping Template Matching Test		
(Template length = 4, Blocks = 2, 4, 8)	0.0314	Passed
The Overlapping Template Matching Test		
(Template length = 4, Blocks = 4, 8)	0.5947	Passed
Maurer's Universal Statistical Test	0.2514	Passed
The Approximate Entropy Test	0.3108	Passed
The Cumulative Sums (CUSUM) Test	0.4512	Passed



(a) (b)

Figure 6: (a) Chaotic sequence for EL map for initial conditions of $r = 1.56$, $x_0 = 0.500000000$, and $c = 2$ and in the same figure, two graphs are plotted for EL map for initial conditions of $r = 1.56$, $x_0 = 0.500000001$, and $c = 2$ and $r = 1.56$, $x_0 = 0.500000002$, and $c = 2$. (b) Chaotic sequence for EL map for initial conditions of $r = 1.5600000000$, $x_0 = 0.5$, and $c = 2$ and in the same figure, two graphs are plotted for EL map for initial conditions of $r = 1.5600000001$, $x_0 = 0.5$, and $c = 2$ and $r = 1.5600000002$, $x_0 = 0.5$, and $c = 2$. It can be seen that initially up to 35 iterations, all the sequences of both graphs are same; however after that point, the sequences are completely out of phase to each other.

$$y_n = k_{n-1}\left(x_n - x_{n-1}\right) + y_{n-1}.$$

$$k_{n-m} = \begin{cases} \dfrac{x_{n-1}}{y_{n-1}}\mu^2 & if\ n < m \\ \dfrac{x_{n-m}}{y_{n-m}}\mu^2 & if\ n \geq m. \end{cases} \tag{8}$$

And the initial seed parameters are

$$\begin{aligned} x_0 &\in [-1, 1], \\ \tan\alpha &\in (-\infty, \infty), \\ \mu &\in (0.05, 1), \\ m &= 2, 3, \ldots, n. \end{aligned} \tag{9}$$

Given these initial parameters, we have

$$\begin{aligned} y_0 &= \mu\sqrt{1 - x_0^2}. \\ k_0' &= -\frac{x_0}{y_0}\mu^2. \\ k_0 &= \frac{\tan\alpha + k_0'}{1 - k_0'\tan\alpha}. \end{aligned} \tag{10}$$

The third chaotic map that will be employed is piecewise linear chaotic map, given as [27]

$$x_{n+1} = \begin{cases} \dfrac{x_n}{p} & if\ x_n \in [0, p], \\ \dfrac{1 - x_n}{1 - p} & if\ x_n \in (p, 1], \end{cases} \tag{11}$$

where $x_0 \in (0, 1)$ and $p \in (0, 1)$ are the initial seed parameters.

3. Proposed Watermarking Algorithm

The proposed work is intended for the online multimedia copyright protection. The watermark is inserted into a carrier based on the decisions made by three chaotic maps. The watermark which we want to insert is taken as a byte of length with eight binary bits, even if it is a digital signal, sound data, image data, video data, or simply text. Beside the watermark insertion, a substitution operation is also performed on watermark to enhance the security of the overall system. The watermarking and inverse watermarking processes are explained as follows.

3.1. Watermarking. The watermarking algorithm in the form of a flowchart is shown in Figure 8. We consider the carrier as a digital image denoted as C with size ro, co, fo, where ro is the number of rows, co is the number of columns, and fo is the number of frames in C; that is, a color image has three frames whereas a gray image has one frame. An image pixel at a specific location of i^{th} row, j^{th} column, and k^{th} frame of that image is denoted as $C(i, j, k)$. The image pixel $C(i, j, k)$ has the range of [0 255] and can be expressed as a binary of 8 bits. We consider the watermark as a digital image too denoted as W with size rw, cw, fw, where rw is the number of rows, cw is the number of columns, and fw is the number of frames in W. Here we consider a two-dimensional watermark image so that we will denote the image pixel of the watermark as $W(i, j)$.

Before the watermark insertion, the watermark is divided into eight binary bits in which 4 Least Significant Bits (LSBs) of each pixel are discarded, keeping the 4 Most Significant Bits (MSBs). The reason for reducing the size of watermark image is to decrease the computational complexity while keeping the texture of watermark image as same as possible; this is

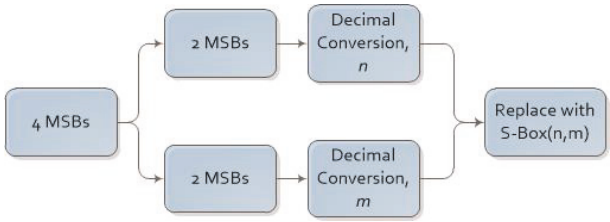


Figure 7: Graphical illustration of 4 × 4 S-Box substitution. The 4 MSBs are divided into two parts, each having two bits. The decimal value of first two bits represents the row number of 4 × 4 S-Box and the decimal value of last two bits represents the column number of 4 × 4 S-Box. The value at that position of S-Box will be replaced with those 4 MSBs.

Table 2: Presentation of S-Box in 4 × 4 matrix.

R/C	0	1	2	3
0	11	1	7	13
1	5	15	4	2
2	6	12	9	14
3	10	0	3	8

shown in the simulated results carried later. The 4 MSBs are substituted with the 4 × 4 S-Box as defined below:

$$W(i,j) = substitution(W(i,j), S-Box) \quad \forall i \in rw,\ \forall j \in cw. \tag{12}$$

For simplicity and convenience, we write substituted watermark image as W as well. The 4 × 4 S-Box is shown in Table 2 and S-Box substitution operation is shown in Figure 7. The 4 MSBs are divided into two parts, each having two bits. The decimal value of first two bits represents the row number of 4 × 4 S-Box and the decimal value of last two bits represents the column number of 4 × 4 S-Box. The value at that position of S-Box will be replaced with those 4 MSBs. After the substitution, the image pixels of W are inserted into C such that the first 2 MSBs are inserted in the left part of C, and the next 2 MSBs are inserted in the right part of C. These MSBs inserted in the positions of left or right parts of C are defined by three chaotic maps; the first chaotic map will define the row number, the second chaotic map will define the column number, and the third chaotic map will define the frame number.

Let x be the chaotic sequence resulted from piece wise chaotic map with initial values of x_0 and p. These initial values are considered as first two secret keys of the proposed algorithm, that is, $k_1 = x_0$ and $k_2 = p$. The length of x is $rw * cw * 100$ and values having under modulo ro. As the chaotic sequence initially has the range of $[0, 1]$, we have amplified that range by multiplying that sequence with 1000 and then limiting the sequence under modulo ro. The x values defined the row positions of C while inserting W in C.

Let x and k be the two chaotic sequences resulting from TD-ERCS chaotic map with initial values of x_0, $\tan\alpha$, μ, and m. For simplicity and convenience, we denote x as y. These four initial values are considered as next four secret keys of the proposed algorithm, that is, $k_3 = x_0$, $k_4 = \tan\alpha$, $k_5 = \mu$, and $k_6 = a$. Moreover, we only need one chaotic sequence from this map; therefore we will only use y sequence. The length of y is $rw * cw * 100$ and values having under modulo $co/2$. As the chaotic sequence initially has the range of [-1, 1], we have amplified that range by first shifting the range from [-1, 1] to [0, 2] and then multiplying that shifted sequence with 1000 and then limiting the sequence under modulo $co/2$. The y values defined the column positions of C while inserting W in C.

Let x be the chaotic sequence resulted from EL chaotic map with initial values of x_0, r, and c. For simplicity and convenience, we denote x as z. These three initial values are considered as next three secret keys of the proposed algorithm, that is, $k_7 = x_0$, $k_8 = r$, and $k_9 = c$. The length of z is $rw * cw * 100$ with under modulo $fo/2$. As the chaotic sequence initially has the range of $[0, 1]$, we have amplified that range by multiplying that sequence with 1000 and then limiting the sequence under modulo $fo/2$. The z values defined the frame positions of C while inserting W in C.

In parallel, the C is divided into C_1 and C_2, where C_1 denotes the left part of C and C_2 denotes the right part of C such that $C_1 = C(:, 1 : co/2, :)$ and $C_2 = C(:, co/2 + 1 : co, :)$. Based on the x, y and z sequences, the image pixels of C_1 and C_2 at $x(i), y(i), z(i)$, i.e., $C_1(x(i), y(i), z(i))$ and $C_2(x(i), y(i), z(i))$, are selected. These image pixels are converted into binary having 8 bits each. The 2 LSBs of these two image pixels are replaced with the 4 MSBs of W. The 2 MSBs of W will be replaced with the 2 LSBs of C_1 at $C_1(x(i), y(i), z(i))$ and the next 2 MSBs of W will be replaced with the 2 LSBs of C_2 at $C_2(x(i), y(i), z(i))$. After the replacement, the binary bits are joined and converted into decimal. This process will continue for all the image pixels present in W. After the replacement of all the image pixels, the two parts are joined together to form a watermarked image, denoted as Wd as explained in Algorithm 1.

3.2. Inverse Watermarking. The inverse watermarking algorithm which is exactly the inverse of the watermarking algorithm in the form of a flowchart is shown in Figure 9. The purpose of inverse watermarking is to extract the watermark to claim the copyright of that digital carrier. The input to the inverse watermarking algorithm is the watermarked image, Wd with size $ro * co * fo$. To have the successful extraction of W, the same set of secret keys will be used.

As a first step of the inverse watermarking algorithm, the chaotic sequences are generated from three chaotic maps using the same set of secret keys. In parallel, for the extraction of watermark, the Wd is first divided into C_1 and C_2, where C_1 denotes the left part of C and C_2 denotes the right part of C such that $C_1 = C(:, 1 : co/2, :)$ and $C_2 = C(:, co/2 + 1 : co, :)$. Based on the x, y and z sequences, the image pixels of C_1 and C_2 at $x(i), y(i), z(i)$, i.e., $C_1(x(i), y(i), z(i))$ and $C_2(x(i), y(i), z(i))$, are selected. These image pixels are converted into binary having 8 bits each. The 2 LSBs of these two image pixels are selected, combined into 4 MSBs, and converted into decimal. This process will continue until all the image pixels present in the W are extracted as shown in

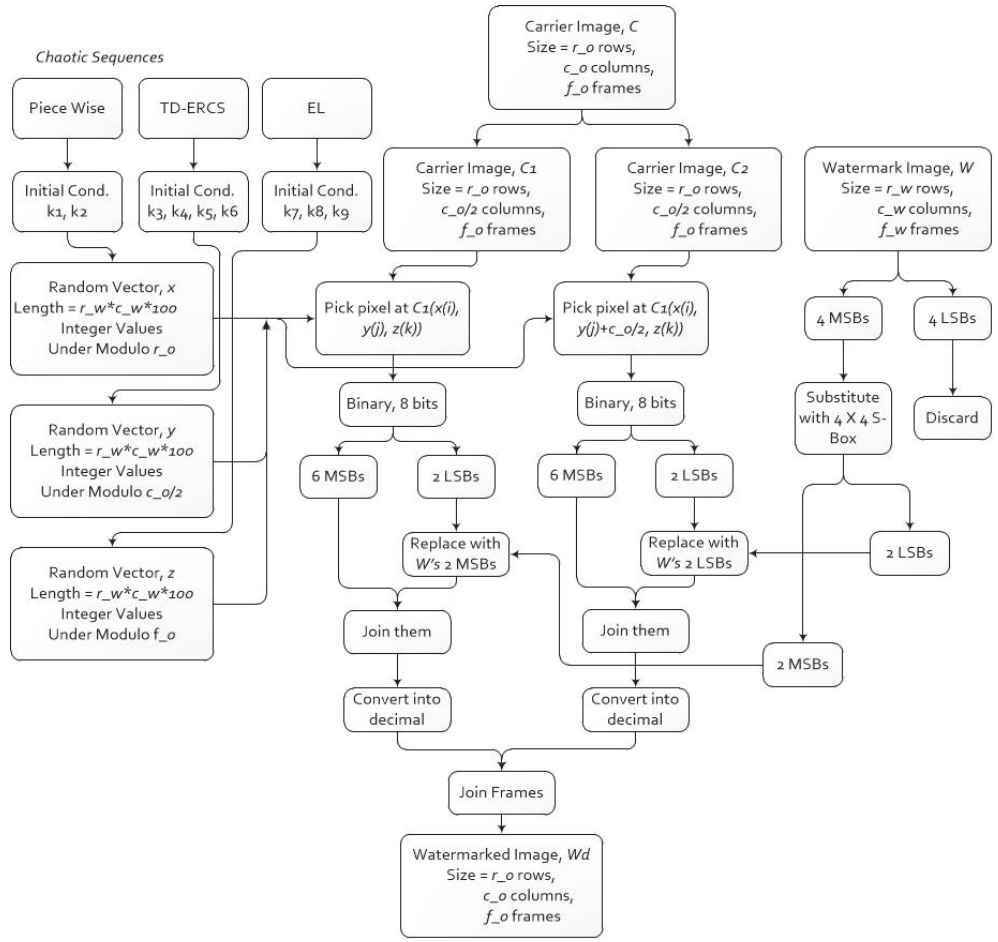


Figure 8: Flowchart of the watermarking algorithm is comprised of three chaotic maps with the help of a 4 × 4 S-Box. For each operation performed with the assistance of a chaotic map, a different set of secret keys is used.

Inputs: Carrier image $C_{(ro\times co\times fo)}$, Watermark image $W_{(rw\times cw)}$, 4 × 4 S-Box, piece wise map, td-ercs map, EL map, and secret keys: $k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9$.
Output: Watermarked image $Wd_{(ro\times co\times fo)}$.

(1) $x = piece\_map(k_1, k_2)$
(2) $y = tdercs\_map(k_3, k_4, k_5, k_6)$
(3) $z = EL\_map(k_7, k_8, k_9)$
(4) $W = substitution(W, S - box)$
(5) $ii = 1, jj = 1, kk = 1.$
(6) $Wd = C$
(7) **for** $i \longleftarrow 1 : rw$ **do**
(8) **for** $j \longleftarrow 1 : cw$ **do**
(9) $W\_a = dec2bin(W(i, j), 8)$
(10) $Wd\_a = dec2bin(C(x(ii), y(jj), z(kk), 8)$
(11) $Wd\_b = dec2bin(C(x(ii), y(jj) + co/2, z(kk), 8)$
(12) $Wd(x(ii), y(jj), z(kk) = bin2dec(Wd\_a(1 : 6), W\_a(1 : 2))$
(13) $Wd(x(ii), y(jj) + co/2, z(kk) = bin2dec(Wd\_b(1 : 6), W\_a(3 : 4))$
(14) $ii = ii + 1, jj = jj + 1, kk = kk + 1.$
(15) **Endfor**
(16) **Endfor**

Algorithm 1: Watermarking procedure is comprised of three chaotic maps with the help of a 4 × 4 S-Box.

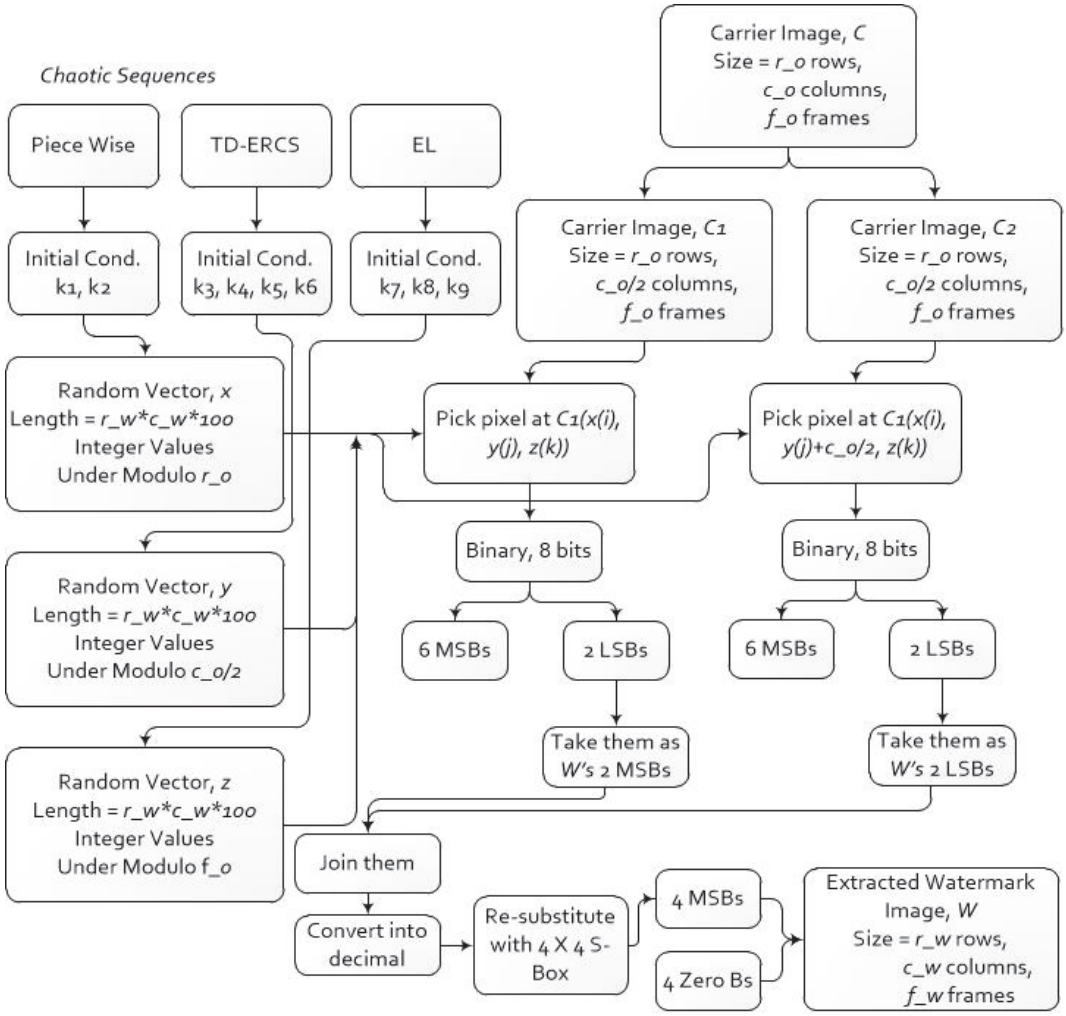


Figure 9: Flowchart of the inverse watermarking algorithm is comprised of three chaotic maps with the help of a 4 × 4 S-Box. For each operation performed with the assistance of a chaotic map, a different set of secret keys is used. However, the secret keys are the same as those used in the watermarking algorithm.

Algorithm 2. As the last step, the image pixels of W are then resubstituted with the same S-Box shown in Table 2 as defined below:

$$W(i,j) = inv_{s}ubstitution\left(W(i,j), S-Box\right) \quad \forall i \in rw,\ \forall j \in cw. \tag{13}$$

4. Simulated Results and Statistical Analysis

The simulations are conducted taking the first baboon as carrier image with size 512 × 512 × 3; i.e., there are 512 rows, 512 columns, and three frames in the baboon color image. We take cameraman image as the watermark with size 225 × 225 × 1; i.e., there are 225 rows, 225 columns, and one frame in the cameraman gray scale image. The size of the watermark image is almost the half of the carrier which is very large; the reason for taking a large watermark is to check the robustness of proposed work. The secret keys which are the initial values of the three chaotic maps employed are listed in Table 3. Before inserting the watermark into the carrier, the watermark is divided into eight binary bits in which 4 Least Significant Bits (LSBs) of each pixel are discarded. The watermark cameraman image is shown in Figure 10(a) and after discarding 4 Least Significant Bits (LSBs) of each pixel, the watermark cameraman image is shown in Figure 10(b). It can be seen that there is not much difference in texture between these two images. Furthermore, we have substituted this after-discarded version of the watermark with the S-Box shown in Table 2. The substituted version of watermark is shown in Figure 10(c). Although the information is visually available in the substituted image, it still provides better security. It is worth mentioning here that we can increase the security by employing multiple S-Boxes instead of a single S-Box but at the cost of extra computational complexity. Before the insertion, the baboon image which is considered as a carrier is shown in Figure 11(a). The watermark which is to

Inputs: Watermarked image $Wd_{(r\_o \times c\_o \times f\_o)}$, 4×4 S-Box, piece wise map, td-ercs map, EL map, and secret keys: $k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9$.
Output: Inverse Watermark image $InvW_{(r\_w \times c\_w)}$.
(1) $x = piece\_map(k_1, k_2)$
(2) $y = tdercs\_map(k_3, k_4, k_5, k_6)$
(3) $z = EL\_map(k_7, k_8, k_9)$
(4) $ii = 1, jj = 1, kk = 1$.
(5) **for** $i \longleftarrow 1 : r\_w$ **do**
(6) **for** $j \longleftarrow 1 : c\_w$ **do**
(7) $InvW\_a = dec2bin(Wdx((ii), y(jj), z(kk)), 8)$
(8) $InvW\_b = dec2bin(Wdx((ii), y(jj) + c\_o/2, z(kk)), 8)$
(9) $InvW = bin2dec(InvW\_a(7:8), InvW\_b(7:8), 0000)$
(10) **Endfor**
(11) **Endfor**
(12) $InvW = inv\_substitution(InvW)$

Algorithm 2: Inverse watermarking procedure is comprised of three chaotic maps with the help of a 4×4 S-Box.

Table 3: Values of secret keys which are the initial conditions of the three chaotic maps employed in the proposed watermarking algorithm.

Maps	Parameters						
	x_0	$\tan\alpha$	μ	m	r	p	c
EL Map	0.5	-	-	-	1.46	-	2
TD-ERCS Map	0.5	1	0.4	50	-	-	-
Piece Wise Map	0.4	-	-	-	3.7	0.9	-

be inserted is shown in Figure 11(b) and after insertion, the watermarked baboon image is shown in Figure 11(c). The histogram of the carrier image is shown in Figure 11(d) and the histogram of watermarked image is shown in Figure 11(e). We can see from the images and histograms of the carrier and watermarked that they are visually similar to each other. After the extraction of the watermark from the watermarked image, we expect to get the cameraman image shown in Figure 11(b). The visual strength of proposed watermarking is further elaborated through the statistical analysis done in the next section.

4.1. Statistical Analysis. To examine the visual strength of the proposed watermarking algorithm, different statistical analyses are performed on carrier and watermark to compare the visual appearance of these two images. The statistical analysis considered in this work is as follows.

4.1.1. Correlation. The correlation of an image is given as [28]

$$Corr. = \sum_{i,j} \frac{(i - \mu i)(j - \mu j)\rho(i,j)}{\varphi_i \varphi_j}, \tag{14}$$

where i, j corresponds to image pixels positions, $\rho(i, j)$ is pixel value at i^{th} row and j^{th} column of image, μ is the variance, and φ is the standard deviation. The correlation analysis determines the similarity between two neighbor image pixels over the whole image having range between $[-1\ 1]$ with 1 showing the perfect correlation.

4.1.2. Entropy. The entropy of an image is given as [28]

$$Entropy = -\sum_{i,j} pr(\rho(i,j)) \log_2 pr(\rho(i,j)), \tag{15}$$

where i, j corresponds to image pixels positions, $\rho(i, j)$ is pixel value at i^{th} row and j^{th} column of image, and $pr(\rho(i, j))$ is the probability of image pixel. Entropy shows the randomness of image having range between [0 8] for an image having 256 gray scales. A greater value of entropy shows the greater amount of randomness.

4.1.3. Contrast. The contrast of an image is given as [28]

$$Contrast = \sum_{i,j} |i - j|^2 \rho(i,j), \tag{16}$$

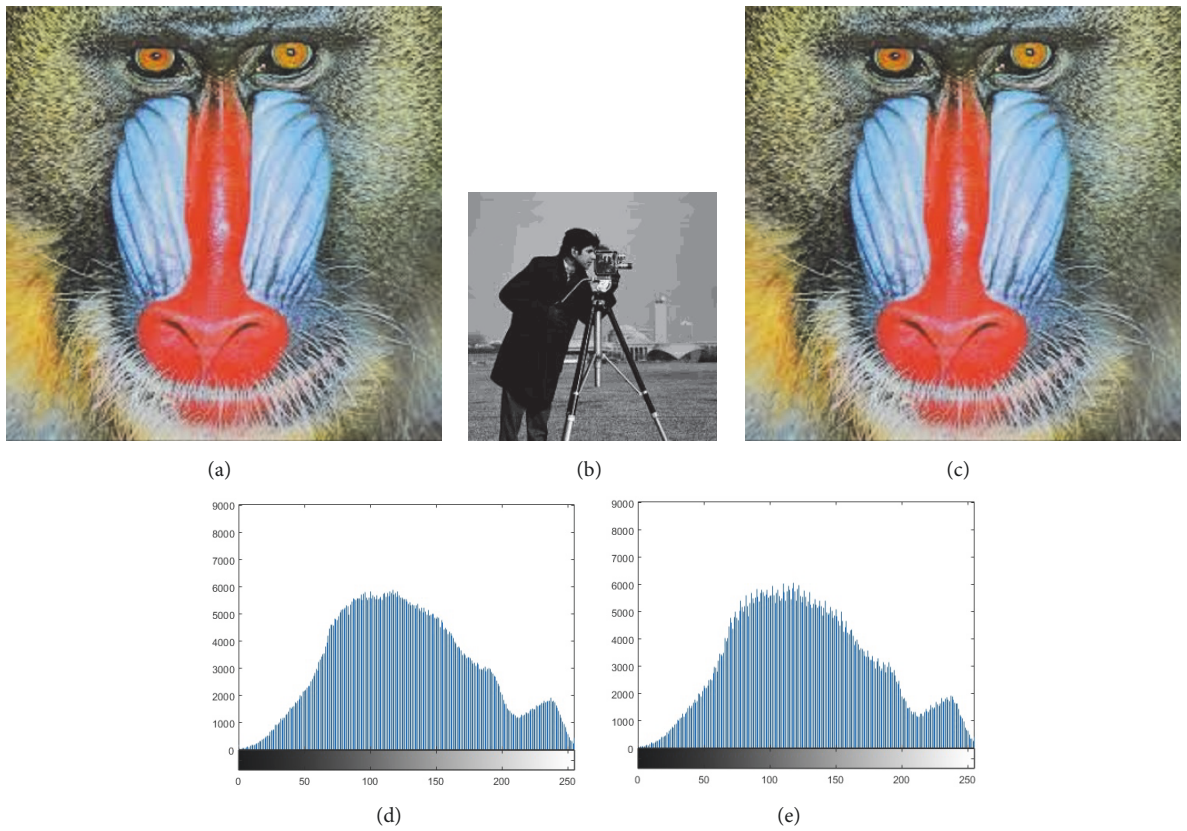
where i, j corresponds to image pixels positions and $\rho(i, j)$ is pixel value at i^{th} row and j^{th} column of image. The contrast analysis of the image enables the viewer to vividly identify the objects in texture of an image. The contrast values ranges from $[0\ (size(Image) - 1)^2]$. The contrast value of a constant image is 0. The greater value of the contrast shows greater variation in image pixels.

4.1.4. Homogeneity. The homogeneity of an image is given as [28]

$$Homo. = \sum_{i,j} \frac{\rho(i,j)}{1 + |i - j|}, \tag{17}$$

(a) (b) (c)

Figure 10: (a) Watermark cameraman image; (b) watermark cameraman image after discarding 4 Least Significant Bits (LSBs) of each pixel. It can be seen that there is not much difference of texture between these two images (a and b). (c) Substituted version of watermark by substituting the after-discarded version of watermark with the S-Box shown in Table 2. Although the information is visually available in substituted image, it still provides better security as compared to (b).



(a) (b) (c)

(d) (e)

Figure 11: (a) The baboon image which is considered as a carrier. (b) The watermark which is to be inserted. (c) After insertion, the watermarked baboon image. We can see that these two images, carrier and watermarked, are visually similar to each other. After the extraction of watermark from the watermarked image, we expect to get the cameraman image shown in (b). (d) Histogram of the carrier image and (e) histogram of the watermarked image.

where i, j corresponds to image pixels positions. The homogeneity analysis processes the closeness of the distribution in the gray level cooccurrence matrix (GLCM) to GLCM diagonal. The range of homogeneity is [0 1].

4.1.5. Energy. The energy of an image is given as [28]

$$Energy = \sum_{i,j} \rho(i,j)^2, \tag{18}$$

Table 4: Comparative statistical analysis on the carrier and watermarked images. The analyses are done on the individual three frames of these two images. It can be seen that, except the entropy analysis, the values of all the other analysis are same for all three frames of these two images showing very good performance.

Frame No.	Images	Analysis				
		Corr.	Entropy	Homo.	Contrast	Energy
1	Carrier	0.9570	7.6605	0.8872	0.2369	0.1062
	Watermark	0.9570	7.6602	0.8872	0.2369	0.1062
2	Carrier	0.9330	7.3575	0.8844	0.2434	0.1244
	Watermark	0.9330	7.3569	0.8844	0.2434	0.1244
3	Carrier	0.9610	7.6779	0.8806	0.2518	0.1040
	Watermark	0.9610	7.6772	0.8806	0.2518	0.1040

Table 5: A comparison of statistical analysis of other watermarking techniques [16–18] with the proposed work applied on different images. It can be seen that the proposed work has superior performance over the other works.

Images	Other Works	Analysis				
		Homo.	Contrast	Energy.	Entropy	Corr.
Pepper	Original	0.8902	0.3311	0.1330	7.5612	0.9207
	Watermarked [16]	0.8917	0.3181	0.1233	7.6003	0.9295
	Watermarked [17]	0.8902	0.3311	0.1330	7.5613	0.9207
	Watermarked [18]	0.8512	0.3241	0.1520	7.4521	0.6241
	Watermarked, Proposed	0.8902	0.3311	0.1330	7.5641	0.9207
Lena	Original	0.8651	0.4141	0.0942	7.7021	0.9444
	Watermarked [16]	0.8687	0.3857	0.1288	7.2512	0.8933
	Watermarked [17]	0.8811	0.3371	0.1130	7.7023	0.9443
	Watermarked [18]	0.9277	0.2688	0.3208	7.6745	0.9688
	Watermarked, Proposed	0.8651	0.4141	0.0942	7.7045	0.9444
Baboon	Original	0.7294	1.0004	0.0817	7.3903	0.6607
	Watermarked [16]	0.8427	0.3531	0.1387	7.1872	0.8933
	Watermarked [17]	0.7294	1.0004	0.0817	7.3903	0.6607
	Watermarked [18]	0.7669	0.7179	0.1028	7.4521	0.6788
	Watermarked, Proposed	0.7294	1.0004	0.0817	7.3945	0.6607

where i, j corresponds to image pixels positions. The energy analysis returns the sum of squared elements in the GLCM. The range of energy is [0 1]. The energy of a constant image is 1.

The results of these analyses considering our proposed algorithm for the carrier and watermarked images are shown in Table 4. The analyses are done on the individual three frames of these two images. It can be seen that except the entropy analysis, the values of all the other analyses are same for all three frames of these two images showing very good performance.

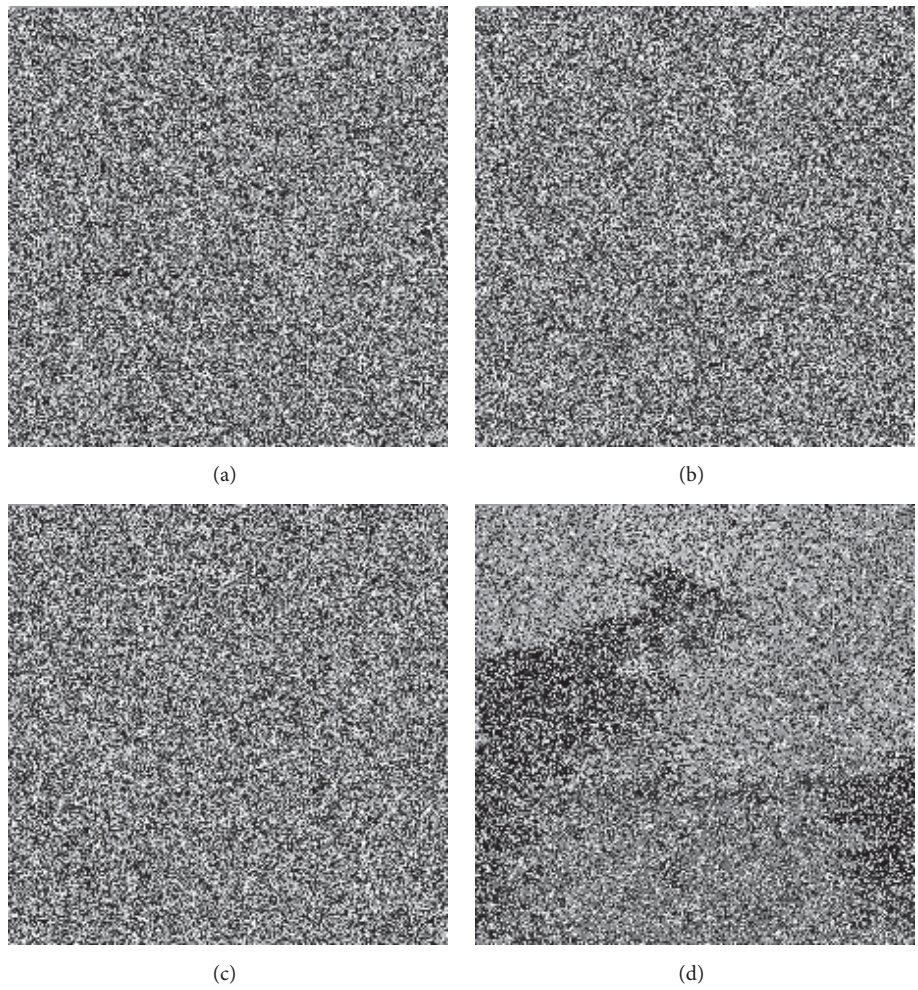
Table 5 presents a comparison of statistical analysis of other watermarking techniques [16–18] with the proposed work applied on different images. It can be seen that the proposed work has superior performance over the other works.

5. Security Analysis

The security analysis assists in determining the strength of any security algorithm. In this section, we have done detailed security analysis which includes key security, noise resistant analysis, and different attacks. These security analyses are described as follows.

5.1. Key Space and Key Sensitivity. Key space refers to the total number of keys that can be used in the watermarking algorithm. We have used initial conditions of three chaotic maps as the secret keys. There are nine secret keys used; the values of these secret keys with their ranges are mentioned earlier. If the average range of a secret key is 10^8, then the total number of different keys that can be used is $10^{8\times9} = 10^{72}$. This is equivalent or more than the 256 binary bits. With this key space, a modern computer will take more than 10^{20} years to check all the combinations.

The key space will only be effective if every key is effective regarding successful extraction of the watermark. For example, key space will only be effective if we tried to extract the watermark from the watermarked image with a slightly change (even change of a single bit) secret key(s) that was used in the embedding of a watermark in carrier image, then the extraction should not be successful. This is known as key sensitivity. In this work, we have embedded the cameraman image with the secret keys mentioned in Table 3

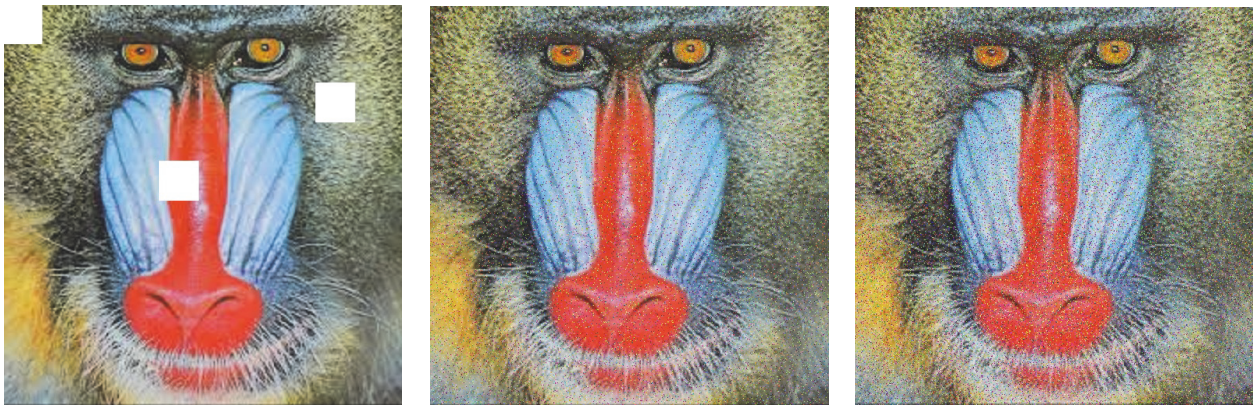


(a) (b) (c) (d)

Figure 12: Four different cases of key sensitivity. (a) Key used in extraction of watermark is changed from $k_1 = x_0 = 0.4$ to $k_1' = x_0 = 0.40000000001$, (b) key used in extraction of watermark is changed from $k_3 = x_0 = 0.5$ to $k_3' = x_0 = 0.50000000001$, (c) key used in extraction of watermark is changed from $k_5 = \mu = 0.4$ to $k_5' = \mu = 0.40000000001$, and (d) key used in extraction of watermark is changed from $k_8 = r = 1.46$ to $k_8' = r = 1.4600001$. In all these images, successful extraction is not done despite a minor change in the secret keys showing the strong results of key sensitivity of our proposed watermarking algorithm.

and watermarked image is shown in Figure 11(c). Then we have tried to extract the watermark image with a slightly change different keys. We have considered four cases. In the first case, we have changed $k_1 = x_0 = 0.4$ to $k_1' = x_0 = 0.40000000001$ while keeping the remaining eight keys as they are. The extracted image with this changed key, k_1', is shown in Figure 12(a); we can see that the extraction is not successful despite a slight change in one of the secret keys. In the second case, we have changed $k_3 = x_0 = 0.5$ to $k_3' = x_0 = 0.50000000001$ and extracted image is shown in Figure 12(b). In the third case, we have changed $k_5 = \mu = 0.4$ to $k_5' = \mu = 0.40000000001$ and extracted image is shown in Figure 12(c). In the fourth case, we have changed $k_8 = r = 1.46$ to $k_8' = r = 1.4600001$ and extracted image is shown in Figure 12(d). In all these images, successful extraction is not done despite a minor change in the secret keys showing the strong results of the key sensitivity of our proposed watermarking algorithm.

5.2. Robustness: Noise Resistant Analysis. One of the necessary features of a modern security system is to be noise resistant [29, 30]. The noise can be added intentionally by an unauthenticated user or can be caused due to the channel noise; channel can be wired or wireless as well. It is a well-known fact that the data that is to be transmitted through any channel is effected by the channel noise. If the watermarked data (image) is corrupted even a little bit, the renown security systems of watermarking do not tend to successfully extract the watermark from the corrupted watermarked image. To handle this situation, error detection and correction are used in parallel to watermarking at transmitter before sending or uploading that watermarked image. When extracting, error correction takes place before the extraction to successfully extract the watermark from the corrupted watermarked image. However, this adds the computational complexity of the whole system. The systems at the transmitter as well as at receiver need more



(a) (b) (c)

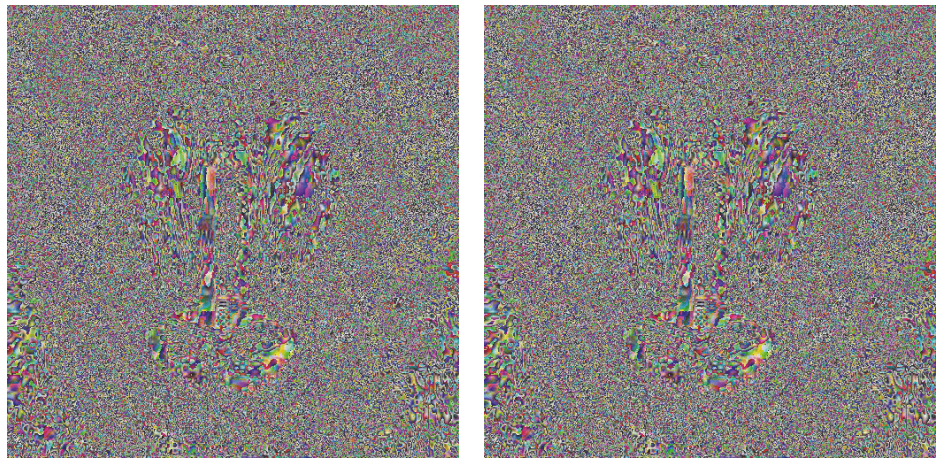
Figure 13: Noise addition results on watermarked images. (a) The watermarked image in which pixels from different locations are corrupted or cropped with the white pixels. (b) The watermarked image in which salt and pepper noise with density equal to 0.05 is added. (c) The watermarked image in which salt and pepper noise with density equal to 0.1 is added to further test the robustness.

(a) (b) (c)

Figure 14: The watermark images after extracting the watermark form the noisy watermarked images. (a) The watermark extracted from Figure 13(a), (b) the watermark extracted from Figure 13(b), and (c) the watermark extracted from Figure 13(c). It can be observed that the extraction is successful with minor changes. The results confirm the robustness of our proposed watermarking algorithm.

time to process the information and therefore need more computational resources. This may not be required in some low profile applications where speed is more required as compared to security. Our proposed watermarking algorithm can extract the watermark from the corrupted watermarked image correctly with some minor changes despite the fact that the watermarked image is undergone through noise addition. We have conducted experiments considering noise addition in watermarked images and then tried to extract the watermarks from those noisy watermarked images. We have embedded the cameraman image in the carrier image of baboon shown in Figure 11(a) with the secret keys mentioned in Table 3 and watermarked image is shown in Figure 11(c). Then we have added different types of noises in it. Figure 13(a) shows the watermarked image in which pixels from different locations are corrupted or cropped with the white pixels. Similarly, we have added salt and pepper noise with a density equal to 0.05. The noisy watermarked image is shown in Figure 13(b). Furthermore, to further test the robustness, we have added salt and pepper noise with a density equal to 0.1. The noisy watermarked image with that density is shown in Figure 13(c). After extracting the watermark form these noisy watermarked images, the watermark images are shown in Figure 14. The watermark extracted from Figure 13(a) is shown in Figure 14(a), the watermark extracted from Figure 13(b) is shown in Figure 14(b) and the watermark extracted from Figure 13(c) is shown in Figure 14(c). It can be observed that the extraction is successful with minor changes. The results confirm the robustness of our proposed watermarking algorithm.

The robustness to noise attacks can be numerically measured as well through the confidence measure proposed



(a) (b)

Figure 15: (a) LSBs of carrier image before the embedding of watermark and (b) LSBs of watermarked image after the embedding of watermark. For the plot, the 4 LSBs of carrier and watermarked images are picked and consider them as 4 MSBs of these two images and for 4 LSBs, 4 zeros are simply considered. It can be seen that there is no visual difference between these two images and thus our proposed algorithm is robust against LSB attack.

Table 6: A comparison results of similarity of different watermarking techniques resulting in applying various noise attacks. Again, it can be seen that the proposed work has superior performance.

Attacks	Other Works	Images/Similarity Index (%)		
		Baboon	Lena	Pepper
Noise	Ref. [17]	72	74	73
	Ref. [16]	72	74	73
	Proposed	88	89	88
Compression	Ref. [17]	67	69	69
	Ref. [16]	67	69	70
	Proposed	85	81	84
Cropping	Ref. [17]	40	42	39
	Ref. [16]	40	42	39
	Proposed	68	69	67

by [31] which returns a numeric value of similarity. The confidence measure is given as [31]

$$Sim = \frac{\sum t_i \cdot s_i}{\sqrt{\sum t_i^2 \cdot \sum s_i^2}} \tag{19}$$

We have performed this confidence measure on our proposed technique as comparison to other works as well. However, for comparison, we have taken the value of similarity as a percentage instead of exact value. Table 6 lists the results of similarity of different watermarking techniques resulting in applying various noise attacks. Again, it can be seen that the proposed work has superior performance.

5.3. LSBs Attack. In this attack, the attacker attempts to find the visual difference between the LSBs of carrier image and watermarked image. As the watermark is embedded into the LSBs of carrier image, this attack has the significance importance. It is assumed that the attacker has access to the watermarked image and carrier image as well (although this assumption leads to the compromise of the carrier and correspondingly the copyrights of this carrier image). The attacker plots the LSBs of the carrier and watermarked images and then tries to find the differences and subsequently tries to extract the watermark. It is required in a good watermarking algorithm that no visual difference should be visible. For the plot, we have picked the 4 LSBs of the carrier and watermarked images and consider them as 4 MSBs of these two images, and for 4 LSBs, we simply consider four zeros. Figure 15(a) shows the LSBs of carrier image before the embedding of watermark and Figure 15(b) shows the LSBs of the watermarked image after the embedding of the watermark. It can be seen that there is no visual difference between these two images and thus our proposed algorithm is robust against LSB attack.

5.4. Adjacent Pixel Difference Analysis. One of the other methods to analyze the LSBs in a watermarked image is to

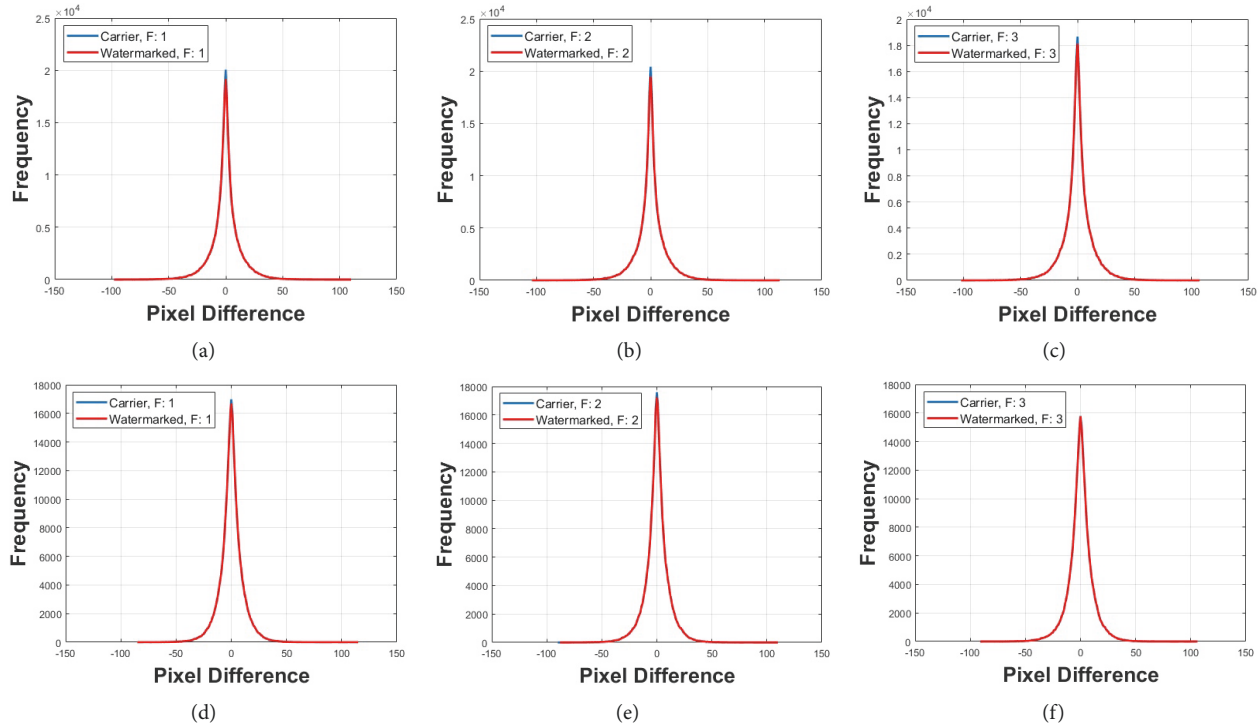
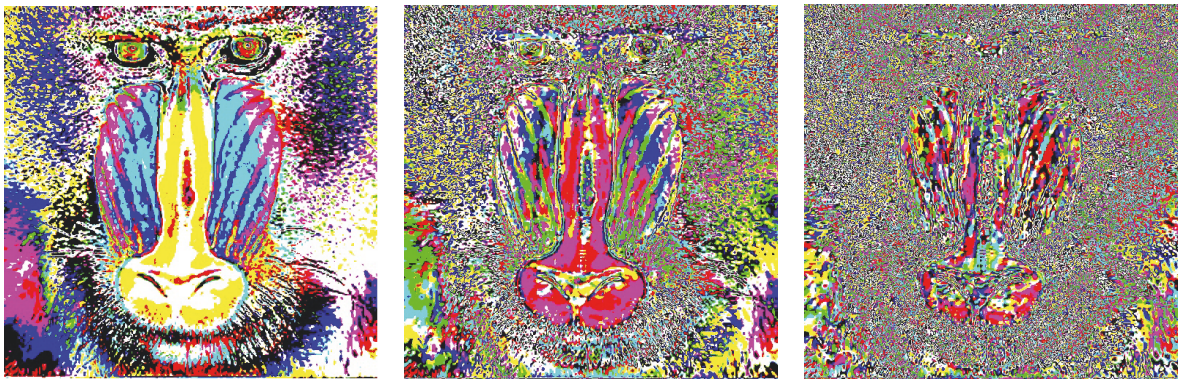


FIGURE 16: (a)-(c) The difference of adjacent pixels for frames 1-3 of carrier and watermarked images when the difference is considered row-wise. In row-wise, the difference of those two image pixels is considered whose positions are in same row but separated by a single column. Similarly, (d)-(f) show the difference of adjacent pixels for frames 1-3 of carrier and watermarked images when the difference is considered column-wise. In column-wise, the difference of those two image pixels is considered whose positions are in same column but separated by a single row. In all these images, the differences of adjacent pixels of carrier and watermarked images are almost the same showing the robustness of proposed watermarking algorithm.

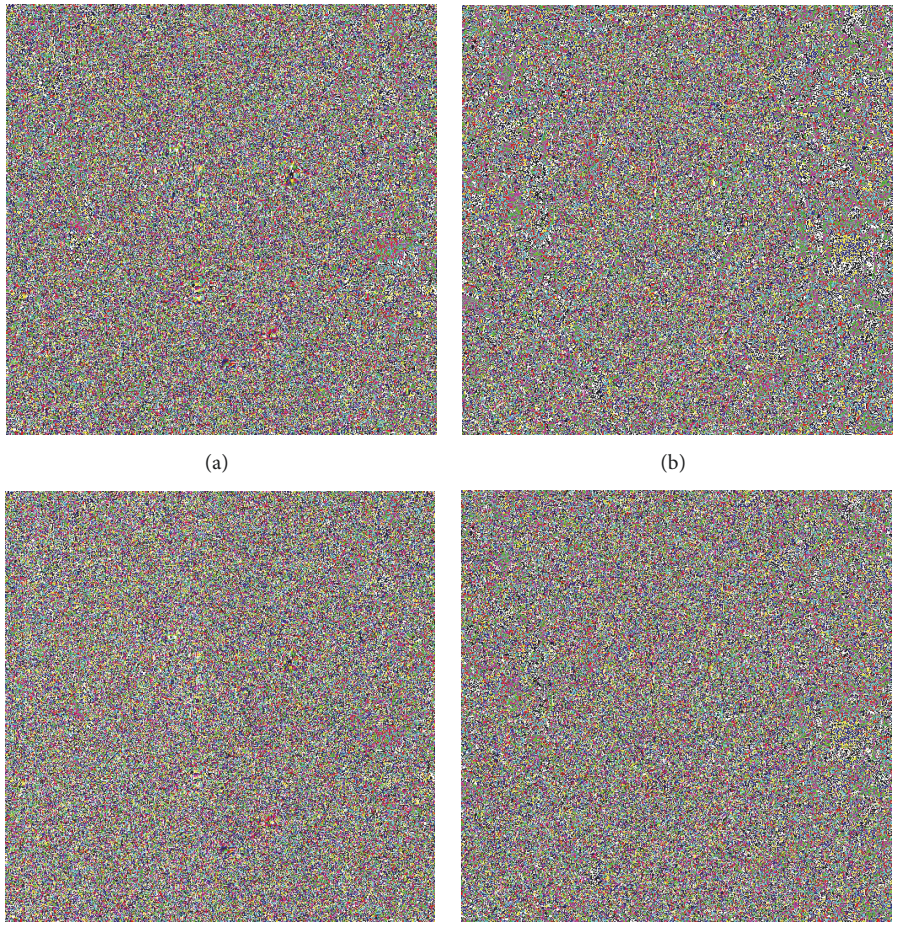
see the difference between adjacent pixels. In an image, the correlation between two adjacent pixels is very high, and thus the difference between these two image pixels is usually close to zero. However, when the watermark is embedded into the LSBs of carrier image to generate a watermarked image, the correlation between adjacent pixels decreases where the watermark is embedded, and the difference of these two image pixels moves away from zero. It is required that the difference between adjacent pixels of both the carrier and watermarked images should be same to each other or near to each other. We have plotted the difference between adjacent pixels of the carrier and watermarked images to see the similarity between these two images. Particularly, we have plotted the difference of adjacent pixels of individual frames of all three frames of the carrier and watermarked images. Figures 16(a)–16(c) show the difference of adjacent pixels for frames 1-3 of the carrier and watermarked images when the difference is considered row-wise. In row-wise, the difference between those two image pixels is considered whose positions are in the same row but separated by a single column. Similarly, Figures 16(d)–16(f) show the difference of adjacent pixels for frames 1-3 of the carrier and watermarked images when the difference is considered column-wise. In column-wise, the difference between those two image pixels is considered whose positions are in the same column but separated by a single row. In all these images, the differences of adjacent pixels of the carrier and watermarked images are almost same showing the robustness of the proposed watermarking algorithm.

5.5. Visual Attack Examining Bit by Bit. It is assumed for a long time since the sophistication of watermarking techniques that the LSBs of a digital image do not contain the important information regarding the image. However, this is not the case for all the images. There are works in the literature on watermarking attacks that suggest examining the LSBs of the carrier and watermarked images to possibly extract the watermark from the watermarked image. To further elaborate this point, we have plotted the single bits of carrier image of a baboon to examine the important information. Figure 17(a) shows the plot of 7th LSB ('xBxxxxxx,' B is the 7th bit) of carrier image; the information is visible due to the high percentage of information (50%) present in it. However, as we move towards the next LSBs, the information tends to lose as can be seen in Figures 17(b) and 17(c) which shows the plot of sixth and 5th LSB, respectively. To show the robustness of proposed watermarking algorithm, we have plotted the single bits of those LSBs of the carrier and watermarked images in which the watermark is embedded, that is, first and second bits. Figure 18(a) shows the plot of second LSB and Figure 18(b) shows the plot of first LSB of carrier image of baboon. As the watermark is embedded in these 2 LSBs



(a) (b) (c)

Figure 17: (a) The plot of 7th LSB ('xBxxxxxx', B is the 7th bit) of carrier image; the information is clearly visible due to high percentage of information (50%) present in it. However as we move towards the next LSBs, the information tends to lose as can be seen in (b) and (c) which show the plot of 6th and 5th LSB, respectively.



(a) (b)

(c) (d)

Figure 18: (a) The plot of second LSB and (b) the plot of first LSB of carrier image of baboon. As the watermark is embedded in these 2 LSBs of carrier image to get the watermarked image therefore it is necessary that the plot of individual bits of first and second LSBs of watermarked image should be similar to the plots of bits of carrier image. (c) The plot of second LSB and (d) the plot of first LSB of watermarked image. It can be seen that these two plots are very similar to the plots of carrier image and they do not give any information about the watermark, thus showing the robustness of proposed watermarking algorithm.

of carrier image to get the watermarked image, therefore, it is necessary that the plot of individual bits of first and second LSBs of the watermarked image should be similar to the plots of bits of carrier image. Figure 18(c) shows the plot of second LSB and Figure 18(d) shows the plot of first LSB of watermarked image. It can be seen that these two plots are very similar to the plots of carrier image and they do not give any information about the watermark thus showing the robustness of the proposed watermarking algorithm.

6. Conclusion

We have developed a unified watermarking algorithm using three different and distinct chaotic maps in which one map is proposed in this work. The embedding of the watermark is operated by the individual chaotic sequence generated by a different chaotic map. The simulation results and security analysis confirmed that the proposed algorithm is secure against well-known attacks. Like all new proposals, we strongly encourage the analysis of our framework before its immediate deployment. The proposed algorithm is a generalized watermarking model that can incorporate changes as required. For instance, the number of substitution boxes can be increased for better security but at the expense of more computational complexity. Furthermore, the work can be extended for the application of steganography as well in which instead of the watermark, the secret message can be inserted for information hiding.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through research groups program under Grant no. R.G.P-1/5/38.

References

[1] Y.-M. Chu, N.-F. Huang, and S.-H. Lin, "Quality of service provision in cloud-based storage system for multimedia delivery," *IEEE Systems Journal*, vol. 8, no. 1, pp. 292–303, 2014.

[2] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, "An efficient approach for the construction of LFT S-boxes using chaotic logistic map," *Nonlinear Dynamics*, vol. 71, no. 1-2, pp. 133–140, 2013.

[3] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical Review Letters*, vol. 64, no. 8, pp. 821–824, 1990.

[4] L. Kocarev, "Chaos-based cryptography: a brief overview," *IEEE Circuits and Systems Magazine*, vol. 1, no. 3, pp. 6–21, 2001.

[5] I. Hussain, A. Anees, M. Aslam, R. Ahmed, and N. Siddiqui, "A noise resistant symmetric key cryptosystem based on S8 S-boxes and chaotic maps," *The European Physical Journal Plus*, vol. 133, no. 4, 2018.

[6] I. Hussain, A. Anees, A. H. AlKhaldi, A. Algarni, and M. Aslam, "Construction of chaotic quantum magnets and matrix Lorenz systems S-boxes and their applications," *Chinese Journal of Physics*, 2018.

[7] I. Hussain, A. Anees, and A. Algarni, "A novel algorithm for thermal image encryption," *Journal of integrative neuroscience*, pp. 1–15, 2018.

[8] A. Anees, A. M. Siddiqui, and F. Ahmed, "Chaotic substitution for highly autocorrelated data in encryption algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 9, pp. 3106–3118, 2014.

[9] I. A. Al-Kadi, "Origins of cryptology: the Arab contributions," *Cryptologia*, vol. 16, no. 2, pp. 97–126, 1992.

[10] T. T. Mapoka, S. J. Shepherd, and R. A. Abd-Alhameed, "A new multiple service key management scheme for secure wireless mobile multicast," *IEEE Transactions on Mobile Computing*, vol. 14, no. 8, pp. 1545–1559, 2015.

[11] A. Anees, W. A. Khan, M. A. Gondal, and I. Hussain, "Application of mean of absolute deviation method for the selection of best nonlinear component based on video encryption," *Zeitschrift fur Naturforschung - Section A Journal of Physical Sciences*, vol. 68, no. 6-7, pp. 479–482, 2013.

[12] H. Liu and X. Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Computers & Mathematics with Applications. An International Journal*, vol. 59, no. 10, pp. 3320–3327, 2010.

[13] A. Anees and Z. Ahmed, "A Technique for Designing Substitution Box Based on Van der Pol Oscillator," *Wireless Personal Communications*, vol. 82, no. 3, pp. 1497–1503, 2015.

[14] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Encryption Standard*, Springer, Berlin, Germany, 2002.

[15] A. Anees and M. A. Gondal, "Construction of Nonlinear Component for Block Cipher Based on One-Dimensional Chaotic Map," *3D Research*, vol. 6, no. 2, 2015.

[16] A. Anees and A. M. Siddiqui, "A technique for digital watermarking in combined spatial and transform domains using chaotic maps," in *Proceedings of the 2013 2nd National Conference on Information Assurance, NCIA 2013*, pp. 119–124, pak, December 2013.

[17] S. S. Jamal, M. U. Khan, and T. Shah, "A Watermarking Technique with Chaotic Fractional S-Box Transformation," *Wireless Personal Communications*, vol. 90, no. 4, pp. 2033–2049, 2016.

[18] S. S. Jamal, T. Shah, and I. Hussain, "An efficient scheme for digital watermarking using chaotic map," *Nonlinear Dynamics*, vol. 73, no. 3, pp. 1469–1474, 2013.

[19] W. Sheng, S. Chen, G. Xiao, J. Mao, and Y. Zheng, "A Biometric Key Generation Method Based on Semisupervised Data Clustering," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 9, pp. 1205–1217, 2015.

[20] M.-H. Lim, A. B. J. Teoh, and K.-A. Toh, "An efficient dynamic reliability-dependent bit allocation for biometric discretization," *Pattern Recognition*, vol. 45, no. 5, pp. 1960–1971, 2012.

[21] W. Sheng, G. Howells, M. Fairhurst, and F. Deravi, "Template-free biometric-key generation by means of fuzzy genetic clustering," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 183–191, 2008.

[22] E. J. Kelkboom, G. G. Molina, J. Breebaart, R. N. Veldhuis, T. A. Kevenaar, and W. Jonker, "Binary Biometrics: An Analytic Framework to Estimate the Performance Curves Under Gaussian Assumption," *IEEE Transactions on Systems, Man, and*

Cybernetics - Part A: Systems and Humans, vol. 40, no. 3, pp. 555–571, 2010.

[23] P. F. Verhulst, "Recherches mathématiques sur la loi daccroissement de la population," *Nouveaux mémoires de l'Académie Royale des Sciences et Belles-Lettres de Bruxelles*, vol. 18, pp. 14–54, 1845.

[24] N. K. Pareek, V. Patidar, and K. K. Sud, "Cryptography using multiple one-dimensional chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 10, no. 7, pp. 715–723, 2005.

[25] A. Rukhin, J. Sota, J. Nechvatal et al., "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Special Publication NIST 800-22, National Institute of Standards and Technology, 2010.

[26] S. L-Yuan, S. K-Hui, and L. C-Bing, "Study of a discrete chaotic system based on tangent-delay for elliptic reflecting cavity and its properties," *Acta Physica Sinica*, vol. 53, no. 9, pp. 2871–2876, 2004.

[27] X. Wang and D. Chen, "A parallel encryption algorithm based on piecewise linear chaotic map," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[28] A. Anees, A. M. Siddiqui, J. Ahmed, and I. Hussain, "A technique for digital steganography using chaotic maps," *Nonlinear Dynamics*, vol. 75, no. 4, pp. 807–816, 2014.

[29] F. Ahmed, A. Anees, V. U. Abbas, and M. Y. Siyal, "A noisy channel tolerant image encryption scheme," *Wireless Personal Communications*, vol. 77, no. 4, pp. 2771–2791, 2014.

[30] F. Ahmed and A. Anees, "Hash-Based Authentication of Digital Images in Noisy Channels," *Robust Image Authentication in the Presence of Noise*, pp. 1–42, 2015.

[31] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.