Alexandria Engineering Journal (2022) 61, 5713–5734

HOSTED BY

ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

An FPP-resistant SVD-based image watermarking scheme based on chaotic control

Wafa' Hamdan Alshoura [a,*], **Zurinahni Zainol** [a,*], **Je Sen Teh** [a,*], **Moatsum Alawida** [a,b,*]

[a] *School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Pulau Pinang, Malaysia*
[b] *Department of Computer Sciences, Abu Dhabi University, Abu Dhabi 59911, United Arab Emirates*

Received 28 July 2021; revised 29 September 2021; accepted 28 October 2021
Available online 12 November 2021

KEYWORDS

Chaotic map;
Image watermark;
Integer wavelet transform;
IWT;
Singular value decomposition SVD

Abstract Image watermarking commonly involves singular value decomposition (SVD) because of its simplicity and minimal effect on image quality. However, SVD-based image watermarking schemes suffer from some drawbacks such as the false positive problem (FPP), and an undesirable trade-off between vital properties such as imperceptibility, embedding capacity, and robustness. To address these drawbacks, we improve upon the SVD-based color image watermarking by incorporating integer wavelet transform (IWT) and chaotic maps. A grayscale image watermark is decomposed into eight gray levels (bit-planes) before being encrypted and re-ordered by a chaotic sequence. Each encrypted bit-plane is inserted into singular values of the sub-bands of a host image. The embedding process is controlled by chaos-based multiple scaling factors (MSF) which are updated in each embedded bit-plane. The resulting values are involved in generating a hash value at the end of the embedding processes. The hash value is used to overcome FPP issues and improves security. Our findings illustrate the proposed scheme's robustness, security, imperceptibility, and capacity. It also exhibits excellent robustness against attacks and its performance surpasses a variety of existing schemes. In addition, the proposed scheme is hypersensitive to even the slightest secret key variations.

1. Introduction

With the evolution of technology and data exchange, there are more digital products around the world. The demand for data protection from illegal copying, editing, distribution, and integration problems has greatly increased. Watermarking schemes are designed to protect property and copying rights of digital products. Digital watermarking embeds or hides special data for the purpose of providing copyright protection for digital media. Digital watermarking schemes commonly have two main

* Corresponding authors at: School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Pulau Pinang, Malaysia.
E-mail addresses: wafahamdan@student.usm.my (W.H. Alshoura), zuri@usm.my (Z. Zainol), jesen_teh@usm.my (J.S. Teh), moatsum.alawida@adu.ac.ae (M. Alawida).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2021.10.052

processes, which are embedding and extraction. The embedding process inserts watermark information in the host digital work such as images, audio, video, code, database, and text. The process can be done using hiding techniques or visual techniques. To protect ownership and copyright, most digital watermarking schemes use hiding technologies. On the other hand, the extraction process is focused on the recovery of the embedded information. Generally speaking, digital watermarking has been used in various applications such as content identification and authentication, fingerprinting, and media file archives [1].

There are two main groups of digital image watermarking schemes: visible and invisible. Visible watermarking schemes embed logos or labels into images that owners want to claim ownership of. Since an image is represented as digital numbers stored as a matrix, it is easy to attack the scheme by removing the inserted information [2]. On the other hand, invisible watermarking schemes have invisible watermarks and are widely used in many applications because the human visual system (HVS) cannot detect and perceive the embedded watermark. Watermarks in invisible methods can be embedded in unknown areas of a host image. All invisible watermarks need to achieve three requirements: robustness, imperceptibility, and security. Robustness refers to the ability to extract ab embedded image from within a watermarked image even when subjected to extreme distortions caused by well-known attacks. The host and watermarked images will have no observable difference. It is also very difficult to discern a particular location, content or to find certain patterns in a watermarked image resulting from the embedding process [3]. Lastly, security refers to the protection of an embedded watermark from malicious adversaries [4,5].

Embedding invisible watermarks can be performed either in the spatial or frequency domain. Spatial invisible watermarks involve modifying the least significant bits (LSB) of 8-bit grayscale pixels. One or two bits of the LSB are modified using watermark information, leading to minimal differences between the host and watermarked images. Spatial invisible watermarks have low computational complexity [6]. However, the embedded watermark is easily extracted or destroyed. The embedding capacity of these schemes is limited and their imperceptibility is controlled by the number of bits that are used for embedding. The frequency domain involves the use of transform methods [7–9]. Discrete cosine transform (DCT) [10], discrete Fourier transform (DFT) [11], discrete wavelet transform (DWT) [12–14], integer wavelet transform (IWT) [15,16] and singular value decomposition (SVD) [17,15,18–22] are transform methods commonly used in digital image watermarking. A host image can be transformed into frequency coefficients from the spatial domain. A watermark can then be inserted into the host by altering these frequency coefficients. Invisible frequency-based watermarks are resistant to attacks but under perform when it comes to imperceptibility. In order to fulfil imperceptibility and robustness requirements, combining more than one transformation method in invisible frequency watermarks has been used [23–26]. Although invisible frequency watermarks are able to resist a myriad of attacks, the imperceptibility requirement limits the embedding capacity. Since the frequency domain watermarks degrade markedly when a large watermark is embedded, hybrid transform methods involving DWT, SVD, DCT and/or IWT have been used to enhance image invisible frequency watermarks [15,27–33].

The rest of this paper is structured as follows: Section 2 introduces a related work, Section 3, provides a brief introduction to SVD, IWT, and chaotic maps. Section 4 then provides details of the proposed scheme which is experimentally evaluated in Section 5. Finally, closing remarks of the paper are available in Section 6.

2. Related work

SVD transform numerically decomposes a digital grayscale image into three matrices, and these matrices hold the digital image content as frequency coefficients. SVD has seen applications in various image watermarking schemes, notably hybrid schemes, as it simplifies the embedding process and is highly stable. Generally, SVD-based watermarking schemes have different embedding processes. They can either embed a watermark image directly or indirectly. In the direct process, a host image is decomposed into three matrices (U_H S_H and V_H). The U_H and V_H matrices are called singular vectors which hold structural contents and are extremely sensitive to minor changes whereas the S_H matrix describes the luminescence of a digital image and consists of singular values. These singular values are commonly used to host watermarks, whereby watermark images are embedded directly under the control of scaling factors to achieve an optimal trade-off between watermarking requirements [34]. The embedding stage will generate what we refer to as side information, later used for the successful recovery of the embedded image [35]. On the other hand, the indirect embedding method involves decomposing the watermark image before embedding the watermark's singular values, S_W into the host image's singular values, S_H [36,19]. Watermark extraction requires the singular vectors, U_W and V_W (side information).

Unfortunately, when the singular values of S_H are used in the embedding process, the false positive problem (FPP) occurs. FPP has three attacks that allow the extraction of fake watermarks by utilizing side information. When an adversary extracts a fake watermark, then he/she claims rightful ownership of the watermarked image (previously host image) [28]. Researchers have introduced several solutions as a means to prevent FPP and other security-related issues. These solutions include the use of hash functions [37], symmetric-key ciphers [36], digital signatures [38], and embedding techniques such as principle component [39,40] and singular vector embedding [15]. All these solutions have successfully achieved a good trade-off between robustness and imperceptibility, two vital watermarking requirements. However, some of the drawbacks of these SVD-based schemes still need to be improved. These drawbacks can be summarized as follows:

- The use of side information as the extraction key still leads to various types of FPP attacks [41,34,42].
- Additional authentication processes based on embedding digital signatures incur computational overhead and are susceptible to different distortions caused by well-known attacks [43,34,44,38,37].
- As the embedded singular vectors, U_W or V_W contain major structural information of the watermark image, they are weak against geometrical and non-geometrical attacks. Minor distortions can adversely affect the extracted image [15,39,40].

- When chaos-based encryption methods are used as a solution, the underlying weaknesses of the chaotic map (e.g. logistic or Arnold transforms) can lead to security issues such as limited keyspace or parameter estimation attacks [45–48].

To obtain desirable trade-offs between robustness and imperceptibility, researchers have used optimization algorithms to identify optimal scale factors. The scale factor is used to control the embedding process through multiplication with the embedded watermark or SVD coefficients. Two types of scale factors have been used, the single scaling factor (SFF) and multiple scaling factors (MSF). MSF is shown to be better than SSF in various watermarking schemes, which achieve the desired results of robustness and imperceptibility [17,15,20,30,21,49]. Larger MSF parameters can lead to high robustness and lower imperceptibility whereas small MSF parameters can lead to lower robustness and higher imperceptibility. Thus, the goal of optimization search algorithms is to find optimal MSF parameters by changing MSF parameters and comparing them with experimental results (statistical tests) in each iteration. After multiple iterations, optimal MSF parameters with a perfect trade-off can be obtained. Several optimization search algorithms have been employed to find the optimal MSF parameters which include genetic algorithms (GA) [49], particle swarm optimization (PSO)[17], multi-objective ant colony optimization (MOACO) [15] and differential evolution (DE) [20,30,21]. However, SVD watermarking based on optimization algorithms have several drawbacks which can be summarized as follows:

- Using optimization search algorithms is computationally expensive as they require multiple iterations to obtain optimal MSF parameters [17,20].
- Optimal MSF parameters are specific to a particular watermarking scheme and cannot be generalized to others [15,50].
- Optimal MSF parameters must be stored in their entirety for successful watermark extraction [49,51].
- Results will vary based on which algorithm has been used for optimization [51,52].

To overcome the drawbacks of existing schemes, we propose a new hybrid SVD-based image watermarking scheme. First, we apply IWT to the host color image. This generates 12 sub-bands from three RGB channels. The watermark grayscale image is decomposed into eight bit-planes. A secret key is used to generate chaotic variables, which are used to generate a chaotic sequence. The bit-planes and sub-bands are encrypted and scrambled by the chaotic sequence. The MSF parameters are produced in an efficient manner using chaotic maps rather than optimization algorithms to manage the embedding level. We then apply SVD on the first eight sub-bands and the corresponding bit-plane is embedded into the singular values of S_i. In each bit plane, MSF parameters are updated for the next embedding process. Upon completion of the embedding process, a hash value is generated by comparing previously MSF parameters and S_i^{New}. The generated hash value can be stored by a trusted third party to protect the host image. The hash value is used during the extraction process to perform authentication, thus addressing FPP issues. One of the goals of the proposed watermarking scheme is to achieve a balanced trade-off between vital watermarking requirements such as imperceptibility, robustness, capacity, and security.. Apart from being highly resistant to attacks, the proposed scheme also supports various MSF parameters and is hypersensitive to secret key changes.

3. Preliminaries

3.1. Singular Value Decomposition (SVD)

In linear algebra, SVD is a numerical decomposition method that separates an orthogonal matrix into independent matrices with their own energy input [39]. A digital image is represented as a matrix, H of size $M \times N$. SVD can be used to decompose the image as follows

$$SVD(H) = U_H S_H V_H = U_H \times S_H \times V_H^T, \tag{1}$$

$$= \begin{vmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,N} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ u_{M,1} & u_{M,2} & \cdots & u_{M,N} \end{vmatrix} \begin{vmatrix} s_{1,1} & 0 & \cdots & 0 \\ 0 & s_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_{M,N} \end{vmatrix} \times \begin{vmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,N} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ v_{M,1} & v_{M,2} & \cdots & v_{M,N} \end{vmatrix} \tag{2}$$

where U_H and V_H are orthogonal matrices of $\mathbb{R}^2$. The left and right matrices refer to the singular vectors which hold the major structural information of a digital image whereas S_H refers to the singular values. These non-negative singular values, ordered in a descending manner, are stored in a diagonal matrix ($S_{1,1} > S_{2,2} > \ldots > S_{N,M}$). The superscript T refers to the transposed matrix.

Due to their desirable properties, SVD-based methods are commonly applied to watermarking schemes to fulfill watermarking requirements and for ease of use. SVD-based image watermarking schemes have the following core advantages:

- The image's singular values S are stable. S values are not altered significantly when a small distortion is applied to the S values or an image.
- Matrix dimensions can either be square or rectangular when decomposed using SVD.
- The singular values represent the luminance of a digital image layer, whereas the image's geometric content is represented by a pair of singular vectors.
- An image and its rotations (either vertically or horizontally) contain a similar amount of non-zero singular values.

3.2. Integer Wavelet Transform (IWT)

The lifting wavelet transform (LWT) was proposed to provide the basic structure for the adaptive wavelets. A classical lifting scheme consists of three basic components: splitting, predicting, and updating. LWT is used in applications such as image compression and image watermarking. Classical LWT

operates on floating-point values. As many applications operate mainly on integers, converting between integers and floating-point values can cause recovery errors due to rounding-off errors. Lifting schemes can also convert integer inputs to integer wavelet transform (IWT) to eliminate such errors. IWT is reversible, so it can be used to prevent recovery errors. IWT can be easily applied to digital images because pixels can inherently be represented by 8-bit integers. Fig. 1 the four sub-bands generated from a single IWT level of the Lena image. The LL has a major image content and is an approximation image, where the LH, HL, and HH high-frequency sub-bands store image details in three directions (horizontal, vertical, and diagonal).

3.3. Chaotic map

Chaos maps have features that make them appealing for many applications such as cryptography and watermarking [53]. A chaos map is a nonlinear dynamical system that can be iterated many times to generate random data sequences with various dimensions [54,55]. Attractive chaotic features include unpredictability, random-like behavior, complexity, dense orbits, and nonlinearity [48]. A chaos map can be one-dimensional (1D) or multi-dimensional (MD). A 1D chaotic map has a single system variable and at least one control parameter in its mathematical definition. All chaotic sequences are produced by iterating the chaotic map starting from the initial system variable (initial condition), with each iteration leading to a new value or a new chaotic point. Each chaotic point is confined to a fixed phase space which varies depending on the map. The initial condition(s) and control parameter(s) are known as the chaotic variables, and the data sequences are also known as the chaotic points. Chaotic maps are known to be highly sensitive to minute variable changes, which lead to different sets of chaotic points being generated [47]. In this paper, simple 1D chaotic maps are used to minimize computational overhead.

In this paper, we use the enhanced logistic and sine chaotic maps that were developed in [56]. The enhanced chaotic maps overcome security issues such as limited chaotic parameter range, short cycle length, and low complexity. The resulting enhanced chaotic maps show better chaotic behaviors compared to classical chaotic maps. The enhanced chaotic maps can be defined as

$$x_{n+1} = (r_1^2 \times x_n \times (1 - r \times x_n) + \frac{r_1}{x_n}) mod 1 \quad (3)$$

$$y_{n+1} = (r_2 \times sin(\pi \times r_2 \times y_n) + \frac{r_2}{y_n}) mod 1 \quad (4)$$

where x_n and y_n are chaotic system variables with a phase space of $(0, 1)$, and r_1 and r_2 are control parameters with a chaotic range of $(0, \infty)$. The enhanced logistic and sine maps are referred to as logistic-G and sine-G respectively.

Chaotic behaviors can be analyzed from a number of viewpoints to illustrate the strength of the enhanced chaotic maps. The bifurcation diagrams depict the relationship between control parameter values and the resulting chaotic points. The bifurcation diagrams of logistic-G and sine-G are shown in Figs. 2 and 3 respectively. The blue shaded region indicates the presence of chaotic behavior within the phase space. A fully shaded region implies that the enhanced chaotic maps have a wide chaotic range. To quantify this chaotic behavior, the Lyapunov exponent can be calculated. Positive values indicate the presence of chaotic behavior which is proportionate to how large the values are. In Fig. 4, The large positive Lyapunov exponent implies hypersensitivity and rapid divergence between chaotic points. Another indicator of chaotic behavior is fuzzy entropy, which calculates the complexity of a chaotic system. A higher value is desirable because it implies high complexity. Fig. 5 shows that the fuzzy entropy for both enhanced chaotic maps is large for the full range of control parameters. Due to these desirable properties, the enhanced chaotic maps were employed to design the proposed watermarking scheme.

4. Proposed scheme

The proposed scheme can be divided into three main phases which include generating the key, embedding, and extracting the watermark. In the first phase, chaotic variables that will be used to iterate chaotic maps are generated from the secret key. The chaotic points are used to create MSF parameters, selecting sub-bands and bit-planes of the host and watermark images. In the embedding and extraction phases, the image watermark is decomposed into eight gray levels and we embed each level into one of the sub-bands of the host color image. 12 sub-bands are generated from a color image after applying IWT transform and we select eight sub-bands for the embedding and extraction phases based on the chaotic points.

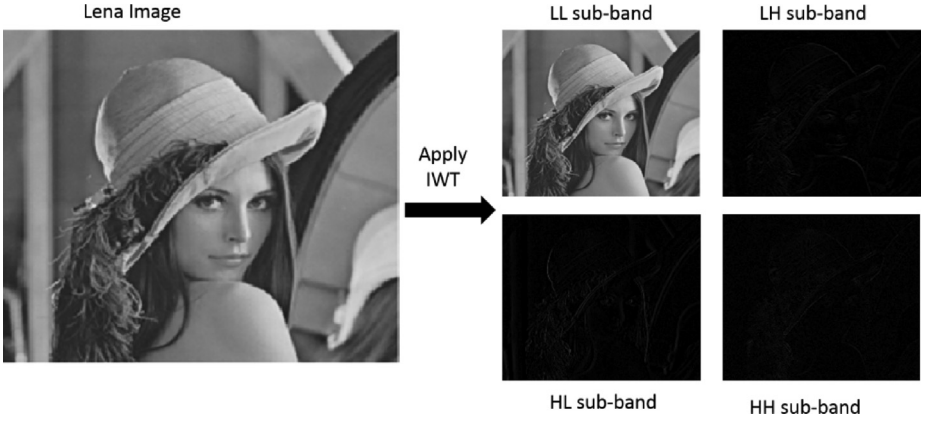

Fig. 1 IWT sub-bands of Lena.

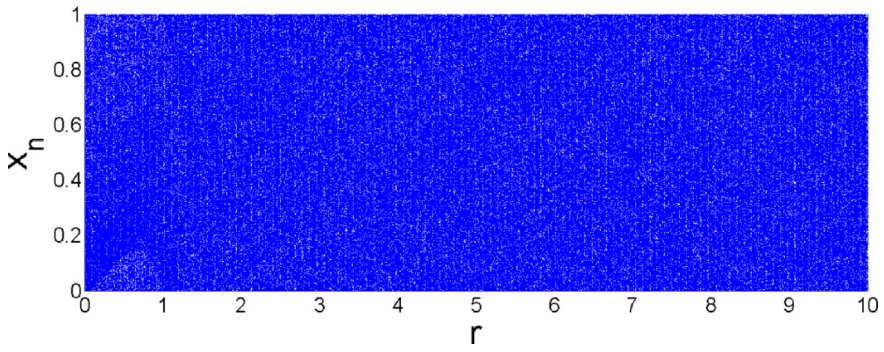

Fig. 2 Bifurcation diagram of the logistic-G map.

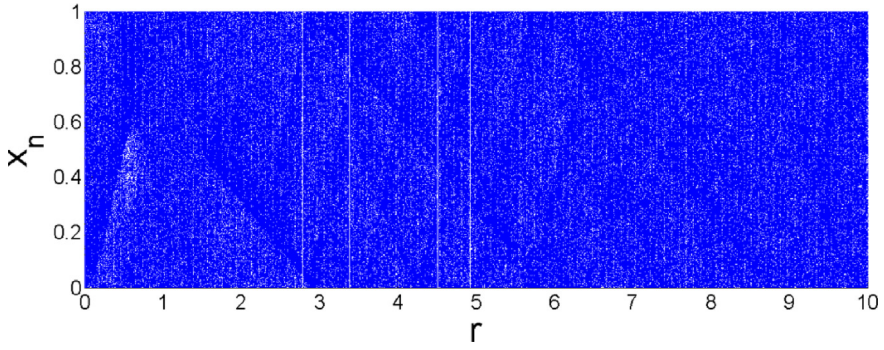

Fig. 3 Bifurcation diagram of the sine-G map.

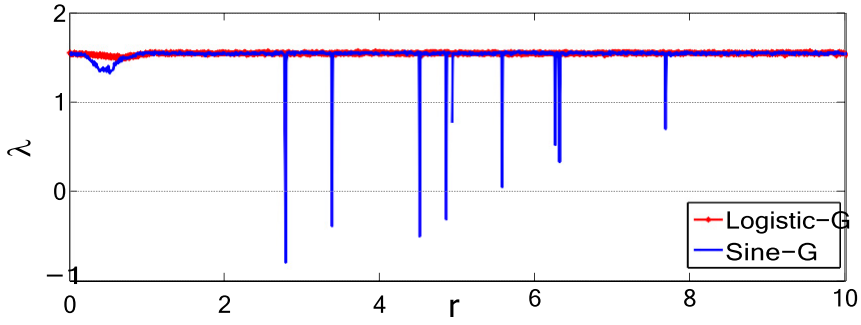

Fig. 4 Lyapunov exponent of the logistic-G and sine-G maps.

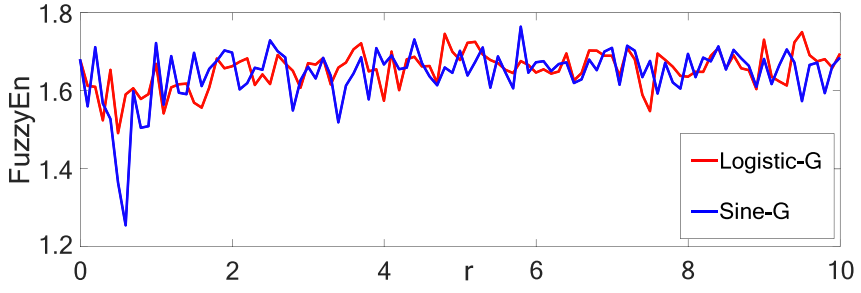

Fig. 5 Fuzzy entropy of the logistic-G and sine-G maps.

Fig. 6 illustrates the general steps involved in the proposed scheme whereas Figs. 7 and 8 depicts the embedding and extraction phases respectively, all of which will be detailed in the following subsections.

The proposed scheme take uses both IWT and chaotic maps in its design. IWT receives integer inputs and transforms them into frequency values that are also integers. As digital images consist of pixels that can be naturally represented as

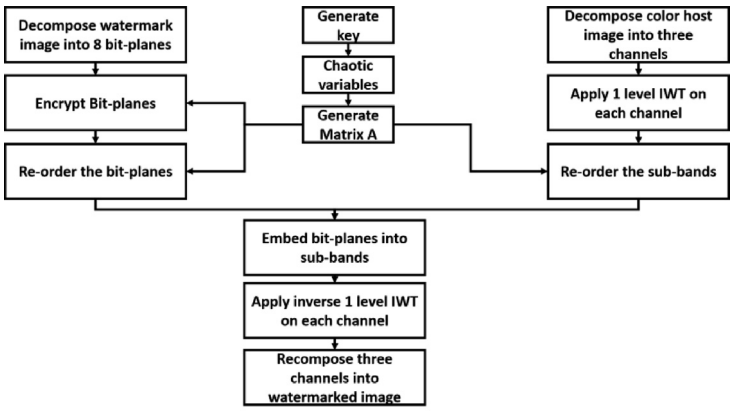

Fig. 6 The flowchart of the proposed scheme (a) Embedding, (b) Extraction.

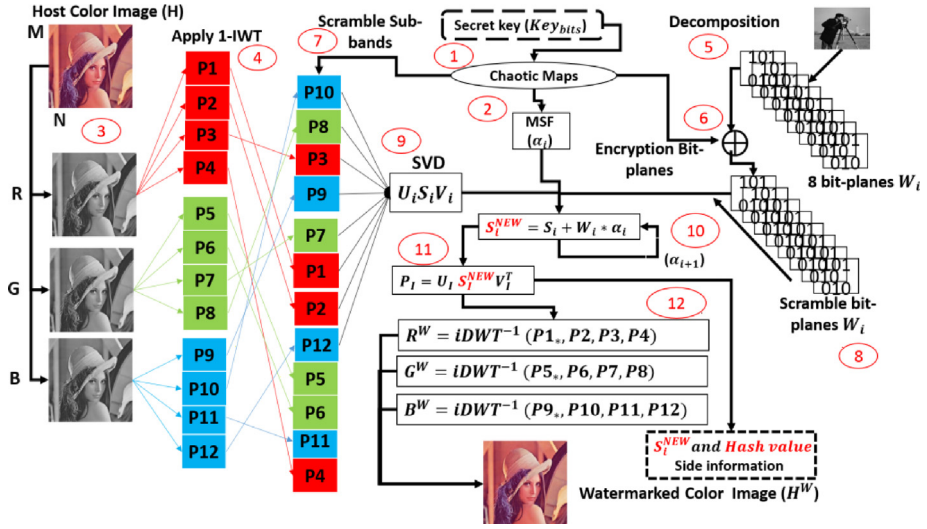

Fig. 7 Embedding process of the proposed scheme.

8-bit integers, IWT is a better option than DWT for transforming a digital image into the frequency domain. As for the inverse transform, round-off errors are removed in IWT, which makes the transformation more accurate than DWT. Chaotic maps contribute to the randomness and nonlinearity of the proposed scheme, providing multiple layers of security to the proposed scheme by producing random MSF parameters, scrambling sub-bands and bit-planes, and encrypting bit-planes of the watermark image. Thus, chaotic maps play a major role in the overall performance and security of the proposed scheme.

4.1. Key generation

Symmetric-key encryption algorithms rely on randomly generated secret keys to protect the confidentiality of sensitive information. A 128-bit secret key is a minimum requirement to resist an exhaustive key search. Also, the encryption algorithm itself must depict a high sensitivity to its encryption key [57]. The proposed scheme uses its secret key to generate chaotic variables for the chaotic maps (logistic-G and sine-G). The 128-bit secret key is converted into chaotic maps parameters

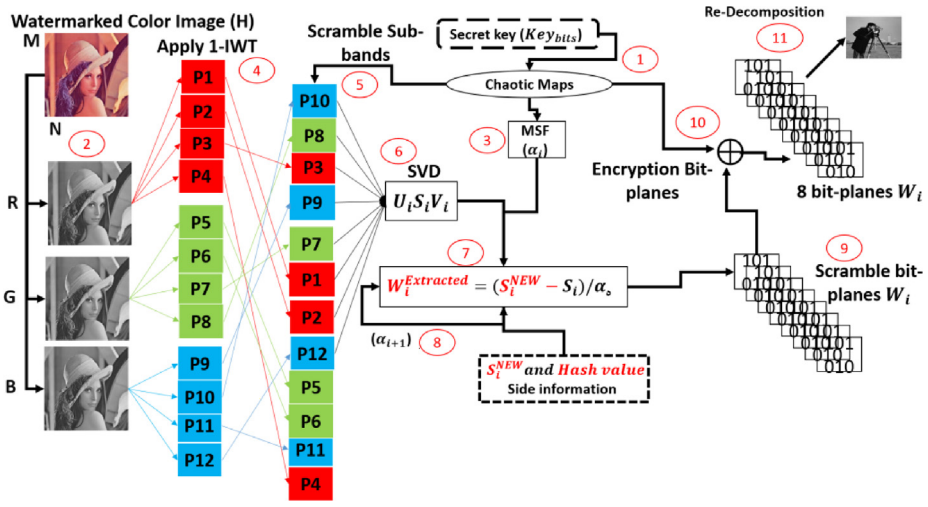

Fig. 8 Extraction process of the proposed scheme.

and initial conditions by using simple mathematical operations. We then iterate the maps to generate a matrix A which controls the embedding process. The entire key generation process includes the following steps:

1. A secret key of 128-bits (key_{bits}) is first selected.
2. The initial conditions x_0 and y_0, and system parameters r_1 and r_2 are calculated based on key_{bits} as

$$x_0 = \sum_{i=1}^{52} \frac{key_{bits}(i)}{2^i}, \tag{5}$$

$$y_0 = \left(\sum_{i=53}^{104} \frac{key_{bits}(i)}{2^{i-52}} + x_0 \right) mod 1, \tag{6}$$

$$r_1 = \left(\sum_{i=1}^{8} key_{bits}(i) \times 2^i + \sum_{i=25}^{76} \frac{key_{bits}(i)}{2^{i-24}} + y_0 \right) mod 10 + 10, \tag{7}$$

$$r_2 = \left(\sum_{i=121}^{128} key_{bits}(i) \times 2^{129-i} + \sum_{i=77}^{128} \frac{key_{bits}(i)}{2^{i-76}} r_1 \right) mod 10 + 10, \tag{8}$$

$$x_0 = (x_0 + r_2) mod 1. \tag{9}$$

3. The initial values for logistic-G and sine-G are $\{x_0, r_1\}$ and $\{y_0, r_2\}$ respectively. In total, the maps undergo $\frac{M}{2} \times \frac{N}{2}$ iterations, and their resulting chaotic sequences are stored in two matrices, X and Y which both have the same dimension of $\frac{M}{2} \times \frac{N}{2}$. These intermediary matrices are used to derive a final chaotic matrix, A as

$$A = \sum_{i=1}^{\frac{M}{2}} \sum_{j=1}^{\frac{N}{2}} \left(((X(i,j) + Y(i,j)) \times 2^{14} mod 1 \right), \tag{10}$$

where i and j denote the rows and columns respectively, while $X(i,j)$ and $Y(i,j)$ are the individual components of the X and Y matrices respectively.

Upon completion of the entire process, A will be made up of random numbers, $r \in (0,1)$. Even a slight modification to the secret key will produce a matrix that differs entirely. This is caused by the sensitivity of the enhanced chaotic maps to parameter changes. A secret key of 128 bits is required to ensure that the keyspace is sufficiently large to overcome brute force attacks. To further expand the key size, the inclusion of more parameters or equations in Step 2 is required. For now, each equation only uses up to 52 bits of the secret key, which is the number of mantissa bits of the double-precision IEEE Standard 754 Floating-Point Numbers.

4.2. Watermark embedding

The embedding process is a simple one that does not need additional transformation methods. It can be summarized as follows:

1. The watermark image is decomposed into eight matrices $W_{1,2,3,4,5,6,7,8}$, which are based on the 8-bit gray levels (bit-planes). Each plane consists of binary values stored within a matrix that has the same dimensions as the original watermark image.
2. Apply IWT transform on the host color image and each channel is decomposed into four sub-bands, leading to a total of 12 sub-bands.
3. The chaotic points are then generated and stored in a matrix A has the same dimension as the watermark image, $\frac{M}{2} \times \frac{N}{2}$. The MSF α_1 is generated as

$$\alpha_1 = (A \times 7^9) mod\, C + 0.0001, \tag{11}$$

where C is a scale value with a fixed range and the constant 7^9 is used to amplify the effect of each value in A. Any other arbitrarily large constant can be used as well. In addition, the constant 0.0001 is used to avoid zero values.

4. The A matrix is converted into a binary matrix β as

$$\beta = dec((A \times 7^9) mod\, 2), \tag{12}$$

where dec is a mathematical function that extracts the decimal portion of a fraction.

5. XOR β with the eight bit-planes to encrypt the watermark image.
6. The 8-values and 12-values of A are selected separately and sorted in ascending order as follows

$$[X, I_X] = sort(A(112, 1:8)) \quad (13)$$
$$[Y, I_Y] = sort(A(20, 1:12))$$

where I_X and I_Y are the index numbers of the values after sorting. E.g. if the number 5 was in position 6 and was sorted to position 1, the first value in I_Y is 6. 112 and 20 refer to the row numbers of A which were selected arbitrarily. Users can select other rows without any detrimental effect on the overall scheme.

7. Select the bit-planes and sub-bands corresponding to the eight values stored in I_X and I_Y. E.g. if the first values are $I_X = 1$ and $I_Y = 6$, we embed them into the first bit-plane and the sixth sub-band of G_{HL}.
8. We apply SVD to the selected sub-band P_i

$$SVD(P_i) = U_{P_i} \times S_{P_i} \times V_{P_i}^T \quad (14)$$

9. We embed values from the matrix W_i into S_{P_i}. The embedding process is defined as

$$S_{P_i}^{new} = S_{P_i} + (W_i) \cdot \alpha_i, \quad (15)$$

where $\cdot$ denotes a dot product operation while α_i refers to the scaling factor.

10. The $S_{P_i}^{new}$ values are then compared with α_i as

$$Comp = \begin{cases} 1 & S_{P_i}^{new} < \alpha_i \\ 0 & Otherwise \end{cases} \quad (16)$$

where $Comp$ is a matrix with the same size as α. The summation of $Comp_i$ is calculated and then use to design the next α_{i+1} as follows

$$\alpha_i = (A \times sum(Comp)) mod\ C + 0.0001 \quad (17)$$

11. We obtain the modified $P_i^{modified}$ sub-band through inverse SVD as

$$P_i^{modified} = U_{P_i} \times S_{P_i}^{new} \times U_{P_i}^T \quad (18)$$

12. Steps 6 to 9 are repeated to embed the eight bit-planes of the watermark into the eight sub-bands of the host color image. In each sub-band and bit-plane, the α_i values are dynamically generated based on the previous α_{i-1} and $S_{P_i}^{new}$ values. This helps to protect the bit-planes against other attacks and binds the proposed scheme together. In the event that an attacker can extract one of the bit-planes, it will be difficult to extract the remaining bit-planes.
13. All $Comp_i$ values are concatenated and used as the hash value of $S_{P_i}^{new}$. When the owner of an image extracts the watermark, the embedded hash value $Hash_E$ is compared to the extracted hash value to determine the validity of the extracted watermark. The hash value can be kept by a TTP and another version can be kept by the owner for security against FPP attacks.
14. We apply an inverse IWT by using $P_i^{modified}$ and the remaining four sub-bands excluded from the embedding process to produce the final watermarked image, H^W.
15. To extract the watermark, we require $Hash_E$, C and $S_{P_i}^{new}$ as the side information in addition to key_{bits} as the secret key.

In total, there are three decomposition operations performed in the proposed scheme. For colored host images, we apply one level of IWT and get 12 sub-bands. For watermark images, we decompose the gray image into 8 bit-planes. The final decomposition is the SVD process for the 8 sub-bands. All of the decomposition operations consist of one level of IWT composition, which allows us to obtain 12 sub-bands from the host image and to prevent embedding into the spatial domain of the host image. For the watermark image, we extract the 8 bit-planes to embed each individual bit, one at a time, rather than larger blocks of 8 bits. This method embeds approximately 50% of the overall watermark image size because only toggled bits (1s) are embedded into the host image, and not 0s. By relying on SVD decomposition, the proposed scheme benefits from all its previously discussed advantages (Section 3.1). Finally, one level of decomposition allows the embedding of different bit-planes into different sub-bands with fewer calculations and reduced errors from high-level decompositions.

4.3. Watermark extraction

Watermark extraction begins by first extracting the eight bit-planes from the watermarked image H^W. These extracted eight bit-planes are used to construct the watermark image. The extracted hash value $Hash_E$ is compared with the extracted hash value $Hash_X$ to confirm that the extracted watermark is correct. The steps involved in the extraction process are as follows:

1. One-level of IWT is applied to the watermarked image, H^W (that may have distortions due to attacks) to produce the 12 sub-bands (each color channel has four sub-bands P_i^W).
2. We then use the secret key to generate chaotic variables which are then used to generate chaotic points based on the same steps in SubSection 4. The resulting matrix A is then used to complete the extraction steps. α_1 with size $\frac{M}{2} \times \frac{N}{2}$ is generated by

$$\alpha_1 = (A \times 7^{9.2}) mod\ C + 0.0001 \quad (19)$$

3. Two indices (I_X and I_Y) are generated by sorting values of A (Eq. 13).
4. Further decompose P_i^W using SVD

$$SVD(P_i^W) = U_{P_i^W} \times S_{P_i^W} \times V_{P_i^W} \quad (20)$$

5. Obtain $W_i^{Extracted}$ by computing

$$W_i^{Extracted} = (S_{P_i}^{new} - S_{P_i^W}) \cdot \alpha_i, \quad (21)$$

where $S_{P_i}^{new}$ is obtained from the side information.

6. The $S_{P_i}^{new}$ is compared with α_i to generate the $Comp$ matrix based on Eq. 16.
7. The summation of $Comp$ is calculated and used to generate the next α_{i+1} based on Eq. 17.
8. The steps (4 to 7) are repeated for the eight rounds and $W_i^{Extracted}$ values are obtained.
9. Sort the gray levels of $W_i^{Extracted}$ back to its natural order $W_{1,2,3,4,5,6,7,8}^{Extracted}$.
10. Decrypt $W_{1,2,3,4,5,6,7,8}^{Extracted}$ by XOR-ing with β that has been generated via Eq. 12.
11. 8-bit values from these extracted bit-planes are mapped to individual image pixels. This is performed based on Algorithm 1

The hash value of the extracted watermark, $Hash_{EX}$ and $Hash_E$ are compared to confirm ownership of the watermark image.

Algorithm 1. Final calculation of the watermark

Data: Extracted bit-planes $W^{Extracted}_{1,2,3,4,5,6,7,8}$, M, N

Result: $W^{Extracted}$, where is the extracted watermark image

$W^{Extracted}$=zeros(M,N);
for *i =1 to 8* **do**
  for *j=1 to M* **do**
    for *k=1 to N* **do**
      if $W^{Extracted}_i == 0$ **then**
        $W^{Extracted}(j,k) = W^{Extracted}(j,k) + W^{Extracted}_i(j,k) \times 2^{i-1}$;
      else
        $W^{Extracted}_i(j,k)$=1;
        $W^{Extracted}(j,k) = W^{Extracted}(j,k) + W^{Extracted}_i(j,k) \times 2^{i-1}$;

4.4. Key management

The hash value $Hash_E$ is kept by a trusted third party (TTP) [24,58–61]. The secret key and side information of the proposed scheme does not need to be kept by the TTP because $Hash_E$ is sufficient to confirm one's ownership of the watermarked image. The hash value is essentially a randomly generated value based on the chaotic points (secret key), the watermark's bit-plane values, and the singular values of the host image's sub-bands. Without the TTP, any adversary can stake a claim to the watermarked image. To overcome this problem, the TTP plays an important role to map each image to its hash value. [24,59].

4.5. Discussion

Our watermarking scheme combines IWT, SVD, and chaotic maps to fulfill various watermarking requirements. The proposed scheme has other advantages that include:

- The secret key, key_{bits} is used to generate the MSF values and select sub-bands and bit-planes that will be used for embedding. The overall security of the proposed scheme is enhanced as it is near-impossible for an adversary to successfully extract the watermark without the secret key.
- The watermark image is decomposed into the eight bit-planes that contain binary values. The imperceptibility and robustness trade-off depends on the 1s in the matrices because the 0s do not have an impact on the embedding process. Rather than embedding MSF values into pixel values, they are embedded into the host image. Thus, 1s do not affect the MSF values. This allows the proposed scheme to resist attacks that introduce distortions. Also, the proposed scheme has high imperceptibility even though the embedding is conducted on eight sub-bands and some of them are the *LL* sub-band.
- In the extraction process, fractions are converted to 1s because any multiplication operations involving 1s will ensure the resulting value remains unaffected. Therefore, our watermarking scheme depicts robustness against well-known attacks.
- α_i is dynamically updated based on the chaotic map and the singular values of the sub-bands of the host color image. Therefore, extracting the watermark image without knowledge of the chaotic points and summation of *Comp* is difficult. Different α_i values make the proposed scheme secure against attacks and achieve an optimal trade-off between the various watermarking requirements without the need for optimization algorithms.
- By employing the matrices of the bit-planes in the embedding process, the overall scheme can support a larger payload (higher capacity).
- FPP issues are tackled by the hash value verification process after the extraction phase. Furthermore, the secret key can protect the watermark image against FPP attacks as it is involved in generating the hash value.
- Chaotic points are used to scramble the 12 sub-bands and eight bit-planes. Changing the chaotic points lead to a change in the sorting of the sub-bands and bit-planes. Furthermore, the encryption of the bit-planes using the chaotic points helps secure the watermark and withstand other attacks.
- Watermark images with a dimension of $\frac{M}{4} \times \frac{N}{4}$ can be embedded by the proposed scheme. To support smaller watermarks, pixels can be duplicated and embedded into other sub-bands, leading to enhanced robustness and capacity.

5. Experimental results and analysis

In the following experiments, five color images are used as the host images (Lena, Peppers, Baboon, Splash, and Airplane (F-16)). These images have a dimension of 512×512 pixels to facilitate a fair comparison with other image watermarking schemes. The Cameraman image (with a dimension of 256×256) was selected as the watermark. These digital images were selected as they are commonly used in many other existing schemes. The host and watermark images are as shown in Fig. 9. All the following experiments and the proposed scheme are coded on MATLAB 2012b and executed on a 64-bit processor and 6 GB RAM. In the proposed scheme, IWT transforms the host color image into 12 sub-bands by relying on the Haar wavelet. IWT allows us to use integers without round-off errors, leading to improved image resolution. After carrying out the IWT transform on the host image, the watermark image is decomposed into eight bit-plans that are embedded into the singular values of the different sub-band. As previously mentioned, the MSFs are computed based on chaotic maps and *C* scale parameters.

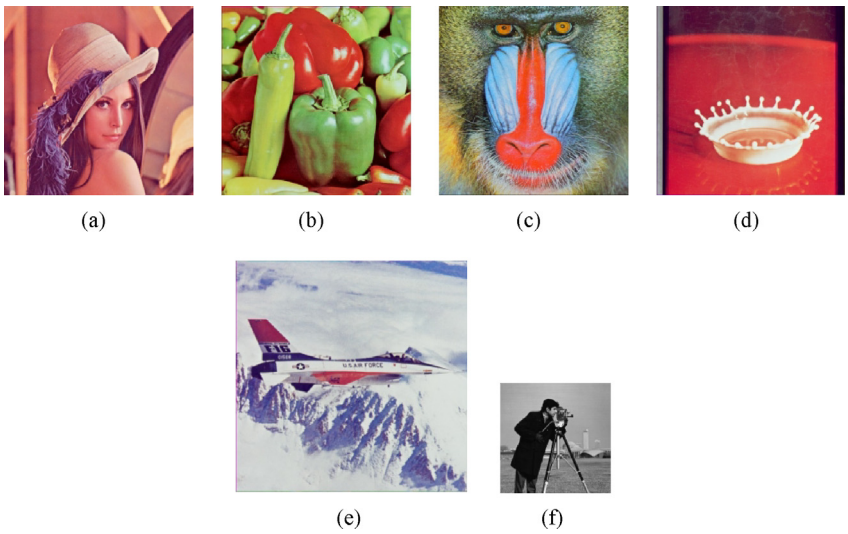

Fig. 9 Host images (a) Lena, (b) Peppers, (c) Baboon, (d) Splash, (e) Airplane (F-16) , (f) the watermark image, Cameraman.

5.1. Imperceptibility and robustness analysis

To assess the imperceptibility and robustness of our watermarking scheme, we calculate the peak-signal-to-noise ratio (PSNR) and normalized correlation (NC) respectively. PSNR is calculated as

$$PSNR = 10\log_{10}\left[\frac{max(H(i,j))^2}{MSE}\right], \tag{22}$$

where $max(H(i,j))$ refers to the host image's largest pixel value. The mean square error (MSE) between the host image, H and the watermarked image, H^W is defined as

$$MSE = \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N}[H(i,j) - H^W(i,j)]^2, \tag{23}$$

where M and N denote the image's rows and columns. Having a high PSNR value implies a minimal difference H and H^W. PSNR is calculated for the host image's three color channels and compared to each other. On the other hand, NC measures the difference between an extracted watermark $W^{Extracted}$ and the original watermark, W. NC is calculated as

$$NC(W, W^{Extracted}) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[W(i,j) - \mu_1][W^{Extracted}(i,j) - \mu_2]}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[W(i,j) - \mu_1]^2} \times \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N}[W^{Extracted}(i,j) - \mu_2]^2}}, \tag{24}$$

where μ_1 and μ_2 are the mean values of W and $W^{Extracted}$ respectively. $NC \approx 1$ implies that both original and extracted watermark images are very similar whereas if $NC = 1$, both images are identical.

To study the imperceptibility, PSNR is calculated for the five host color images underline different MSF intervals, the results of which are tabulated in Table 1. We can observe that the proposed scheme can embed the watermark image with high PSNR values for various MSF intervals because the bit-planes are used in the embedding process and have a low effect on the digital image. Small or large MSF intervals minimally affect the PSNR values. Thus, the proposed embedding strategy does not require the use of any optimization algorithms to identify optimal MSF values. Next, a watermarking scheme is considered robust if the extracted watermark is nearly identical to the original under various adverse conditions. Table 2 contains the proposed scheme's NC results based on various MSF intervals. The NC values depict the scheme's high robustness and flexibility. Results show that the watermark can be successfully extracted by the proposed watermarking scheme regardless of the size of the MSF intervals, with NC values that are close to 1.

Next, we calculate the NC values for the Lena and Peppers host color images under geometrical and non-geometrical attacks. These NC values are compared to another existing scheme in Tables 3 and 4. Also, five MSF intervals were used in this experiment to study the robustness based on two factors (different attacks and different MSF parameters). We used 15 various geometrical and non-geometrical modifications (distortions or attacks) on the color watermarked images. The attacks are performed on the three channels (R, G, and B) before extracting the embedded watermark. As shown in both tables, the proposed scheme can successfully extract the embedded watermark images with high NC values in the presence of distortions regardless of the MSF intervals. The NC values are near-ideal for all the different attacks because the embedding is done based on the bit-planes. As each plane consists of 1s and 0s, close to half of the plane values do not have any effect on the embedding process. Furthermore, the sub-bands for the embedding process were selected randomly based on the chaotic points and some of them may be LL sub-bands that are robust against attacks.

For watermarking schemes where optimization algorithms are used, the MSF parameters are generated after multiple iterations based on a fitness function. Many iterations are required to achieve a desirable trade-off. In this scheme, the

MSF parameters are generated by chaotic maps which are random and more efficient. The MSF parameters are also dependent on the secret key and do not need to be stored as side information. In addition, this scheme generates eight MSF matrices based on the chaotic matrix and *Comp* values generated while embedding into the bit-planes. Thus, the MSF range can either be small or large values while still achieving high imperceptibility and robustness against attacks. Apart from having improved PSNR and NC results, the proposed method of producing different MSF values is also more flexible and faster than relying on optimization algorithms. Fig. 10 depicts the extracted watermark images obtained from watermarked color images that have been subjected to attacks. The results show that the watermarks were extracted successfully with minimal distortion.

The proposed scheme is compared to other existing schemes in terms of imperceptibility and robustness using various host color images. The PSNR imperceptibility values for the proposed scheme and other schemes are listed in Table 5. Other existing schemes [15,49] have PSNR values that are lower than the proposed scheme, implying that the proposed scheme has improved imperceptibility. Table 6 provides further imperceptibility comparisons between the proposed scheme and other recently proposed schemes, showcasing that the proposed scheme has higher quality when embedding 8 bit-planes into the host image. This provides empirical evidence that the proposed scheme has less distortion and an adversary cannot extract information from the watermarked image.

In addition, Table 7 compares the NC robustness values for various schemes under seven well-known attacks. The NC results of the proposed scheme are near-ideal as compared to the other schemes, which indicates successful watermark extraction with minimal distortions. This can be attributed to how the proposed scheme performs watermark extraction using bit-planes (which contain either fractional or zero values) rather than frequency coefficients or the pixel values that decrease the precision of the recovered pixels.

Table 8 shows the NC values for three critical attacks (Salt and pepper, Gaussian noise, and Speckle noise). The NC results of the proposed scheme outperform the other existing schemes, and the embedded watermark can be extracted with lower distortions. This is due to how the bit-planes of the watermark image are embedded into the singular values of the different sub-bands of the host color image rather than centralized embedding into only one sub-band. In addition, recovering 1 or 0 values is more efficient than fractions or integers. Tables 9–13 illustrate some of the attacks with different types of noise and densities. Two existing schemes [62,34] are used for comparison. Results show that the proposed scheme has excellent NC values under different noise density levels as compared to the two prior schemes.

5.2. FPP analysis

The proposed scheme is suitable for the copyright protection of digital images because it can overcome FPP issues by comparing extracted hash values with the ones stored by a TTP. In addition to side information such as C and $S_{P_i}^{new}$, the secret key key_{bits} is also used to embed and extract the watermark. In other existing schemes, side information plays a bigger role during extraction as compared to the secret key, making these schemes susceptible to other attacks and modifications. The

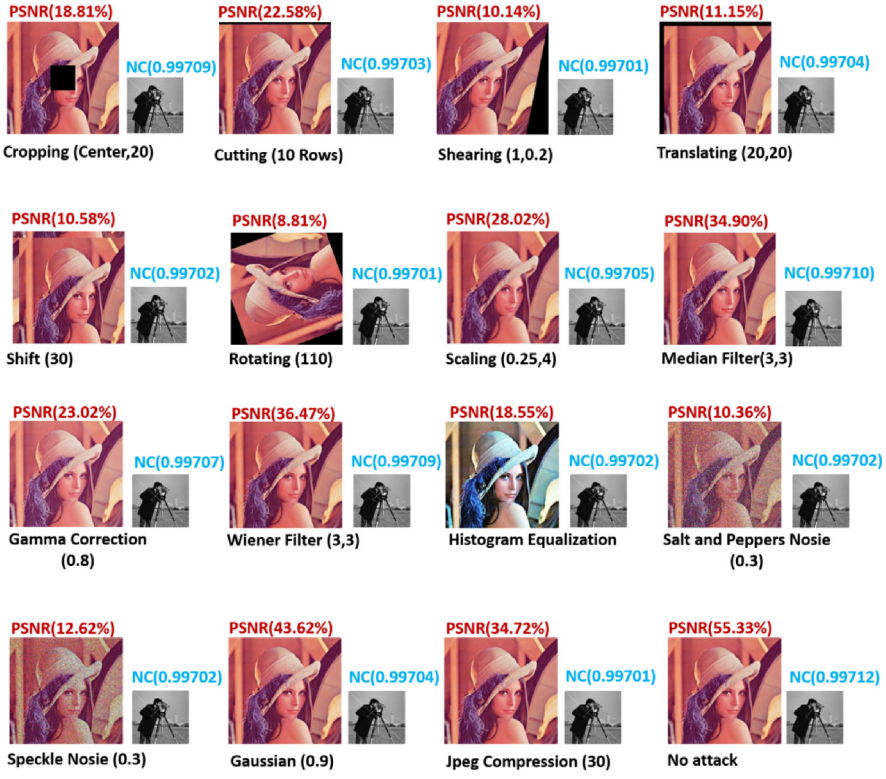

Fig. 10 Watermarked image and extracted watermark against different attacks.

Table 1 Imperceptibility (PSNR) results for different MSF parameters.

Test image	RGB channel	MSF				
		[0.0001 – 0.01]	[0.0001 – 0.1]	[0.0001 – 0.5]	[0.0001 – 1]	[0.0001 – 5]
Lena	R	57.76	57.74	47.45	56.43	49.70
	G	56.10	56.09	55.87	54.92	46.31
	B	53.07	53.06	52.89	52.26	45.64
Peppers	R	52.81	52.81	52.69	52.27	45.95
	G	54.53	54.55	54.41	53.64	45.09
	B	54.26	54.25	54.08	53.32	45.13
Baboon	R	55.75	55.61	55.46	54.86	48.35
	G	54.24	54.15	54.07	53.69	47.38
	B	54.84	54.64	54.51	53.97	48.12
Splash	R	58.89	58.84	58.49	57.39	50.47
	G	54.84	54.77	54.65	53.93	45.34
	B	55.64	55.45	55.20	54.42	47.65
Airplane (F-16)	R	53.42	53.32	52.14	52.36	50.47
	G	54.26	54.18	54.01	53.21	45.34
	B	53.75	53.10	52.94	52.19	47.65

Table 2 Robustness (NC) results for various MSF parameters without attacks.

Test image	CMSF				
	[0.0001 – 0.01]	[0.0001 – 0.1]	[0.0001 – 0.5]	[0.0001 – 1]	[0.0001 – 5]
Lena	0.99712	0.99717	0.99732	0.99734	0.99745
Peppers	0.99714	0.99742	0.99712	0.99788	0.99789
Baboon	0.99744	0.99799	0.99754	0.99761	0.99784
Splash	0.99762	0.99765	0.99779	0.99781	0.99784
Airplane (F-16)	0.99703	0.99769	0.99753	0.99767	0.99777

Table 3 Robustness (NC) results for different MSF parameters under different attacks on the Lena image.

Attacks	MSF					Ref.[15]
	[0.0001 – 0.01]	[0.0001 – 0.1]	[0.0001 – 0.5]	[0.0001 – 1]	[0.0001 – 5]	
Cropping (Center, 20)	0.99709	0.99713	0.99721	0.99713	0.99712	0.9200
Cutting (10 rows)	0.99703	0.99711	0.99712	0.99711	0.99711	0.9822
Shearing (1, 0.2)	0.99701	0.99711	0.99722	0.99721	0.99724	0.9018
Translating (20, 20)	0.99704	0.99714	0.99715	0.99721	0.99734	0.9380
Shifting (30)	0.99702	0.99715	0.99730	0.99729	0.99732	0.9907
Rotating (110)	0.99701	0.99707	0.99712	0.99712	0.99733	0.9479
Scaling (0.25, 4)	0.99705	0.99709	0.99714	0.99718	0.99721	0.9680
Median filter (3, 3)	0.99710	0.99705	0.99721	0.99724	0.99725	0.9974
Gamma Correction (0.8)	0.99707	0.99713	0.99724	0.99722	0.99724	0.9939
Wiener Filter (3, 3)	0.99709	0.99714	0.99721	0.99724	0.99734	0.9901
Histogram Equalization	0.99702	0.99112	0.99324	0.99244	0.99422	0.9311
Salt Peppers Noise (0.3)	0.99702	0.99709	0.99712	0.99731	0.99738	0.9353
Speckle Noise (0.3)	0.99702	0.98714	0.99721	0.99728	0.99731	0.9152
Gaussian Noise (0.3)	0.99704	0.99714	0.99715	0.99722	0.99723	0.9003
JPEG Compression (30)	0.99701	0.99708	0.99713	0.99721	0.99724	0.9930

Table 4 Robustness (NC) results for different MSF parameters under different attacks on the Pepper image.

Attacks	CMSF					Ref.[15]
	[0.0001 − 0.01]	[0.0001 − 0.1]	[0.0001 − 0.5]	[0.0001 − 1]	[0.0001 − 5]	
Cropping (Center, 20)	0.99685	0.99694	0.99701	0.9714	0.99719	0.9340
Cutting (10 rows)	0.99654	0.99665	0.99701	0.99748	0.99749	0.9757
Shearing (1, 0.2)	0.99642	0.99665	0.99685	0.99715	0.99724	0.9130
Translating (20, 20)	0.99635	0.99624	0.99689	0.99714	0.99745	0.9470
Shifting (30)	0.99689	0.99692	0.99714	0.99722	0.99729	0.9910
Rotating (110)	0.99684	0.99703	0.99714	0.99735	0.99754	0.9507
Scaling (0.25, 4)	0.99623	0.99654	0.99684	0.99704	0.99731	0.9720
Median filter (3, 3)	0.99664	0.99718	0.99718	0.99753	0.99771	0.9969
Gamma Correction (0.8)	0.99211	0.99315	0.99335	0.99121	0.99245	0.9956
Wiener Filter (3, 3)	0.99684	0.99692	0.99699	0.99701	0.99724	0.9892
Histogram Equalization	0.99621	0.99621	0.99612	0.99714	0.99719	0.9405
Salt Peppers Noise (0.3)	0.99654	0.99661	0.99665	0.99672	0.99701	0.9433
Speckle Noise (0.3)	0.99622	0.99623	0.99628	0.99636	0.99646	0.9277
Gaussian Noise (0.3)	0.99675	0.99688	0.99698	0.99701	0.99708	0.9878
Jpeg Compression (30)	0.99687	0.99689	0.99704	0.99709	0.99731	0.9965

Table 5 Imperceptibility comparison between various schemes.

Test Image	Proposed scheme	Ref.[15]	Ref.[49]
Lena	**57.76**	42.9245	39.56
Peppers	**52.81**	42.9477	39.98
Baboon	**55.75**	42.9159	39.31

Table 6 Imperceptibility (PSNR) results for other schemes.

Scheme	PSNR
Proposed scheme	57.76
Ref. [63]	57.41
Ref. [64]	55,85
Ref. [13]	44.19
Ref. [7]	52.53

secret key has significant characteristics that support the watermarking scheme to protect the embedded watermark during the extraction process such as sensitivity to slight changes. Unfortunately, side information itself is not sensitive to slight modifications and can be leveraged by adversaries for fraudulent ownership claims. In the proposed scheme, the secret key is used to control all embedding steps. Therefore, changes to even a single bit of the secret key will lead to a large change to the embedding results, especially the resulting hash value. Figs. 11–13 illustrate how the proposed scheme overcomes three attacks. We now analyze the robustness of the proposed scheme based on the three FPP scenarios:

- **FPP Scenario 1:** We denote the Lena image, Cameraman and Woman images as L, C and W respectively. L is used as the host image while C and W are watermarks. We embed C into L (version 1 of Lena) to generate a watermarked image, LC. Next, W is embedded into another copy of L (version 2 of Lena) separately to obtain another watermarked image, LW. The secret key and the corresponding

Table 7 Robustness (NC) comparison under different attacks between various schemes.

Attacks	Ref.[65]	Ref.[20]	Ref.[21]	Ref.[42]	Proposed scheme
Histogram Equalization	0.9934	0.9849	0.9664	**0.9982**	0.99702
Gaussian Noise	0.9849	0.9244	0.9358	0.9567	**0.99704**
JPEG Compression	**0.9991**	0.9954	0.9875	0.9942	0.99701
Gamma Correction	**0.9981**	0.9952	0.9585	0.9948	0.99707
Median filter	0.9932	0.9894	0.9458	0.9903	**0.99710**
Cropping	0.9907	0.9592	0.5789	0.9878	**0.99709**
Shifting	0.9934	0.9899	0.5435	0.9847	**0.99702**

Table 8 Comparing NC values of the proposed schemes with other schemes.

Attacks	NC values Ref. [66]	NC values Ref.[67]	NC values Ref.[68]	NC values Ref.[69]	NC values by proposed scheme
Salt and Pepper	0.9894	0.894	0.9736	0.9734	**0.99702**
Gaussian Noise	0.9762	0.969	0.9849	0.9841	**0.99704**
Speckle Noise	**0.99702**	0.989	0.9522	0.9275	0.99701

Table 9 Salt and Pepper different noise attacks.

Attack	NC		
	Proposed Scheme	Ref. [62]	Ref. [34]
Salat and Pepper 0.5	**0.99675**	0.6320	0.5870
Salat and Pepper 0.3	**0.99702**	0.6650	0.6800
Salat and Pepper 0.1	**0.99712**	0.7830	0.9460
Salat and Pepper 0.01	**0.99714**	0.9510	0.9710
Salat and Pepper 0.001	**0.99721**	0.9750	0.9950

Table 10 Speckle noise attacks.

Attack	NC		
	Proposed Scheme	Ref. [62]	Ref. [34]
Speckle Noise 0.5	**0.99768**	0.6900	0.7020
Speckle Noise 0.3	**0.99702**	0.7190	0.7560
Speckle Noise 0.1	**0.99711**	0.8100	0.8740
Speckle Noise 0.01	**0.99714**	0.9510	0.9720
Speckle Noise 0.001	**0.99718**	0.9720	0.9940

Table 11 Gaussian noise attacks.

Attack	NC		
	Proposed Scheme	Ref. [62]	Ref. [34]
Gaussian Noise 0.5	**0.99684**	0.6380	0.5890
Gaussian Noise 0.3	**0.99704**	0.6520	0.6190
Gaussian Noise 0.1	**0.99706**	0.7050	0.7400
Gaussian Noise 0.01	**0.99709**	0.8780	0.9360
Gaussian Noise 0.001	**0.99711**	0.9790	0.9820

Table 12 Median filter attacks.

Attack	NC		
	Proposed Scheme	Ref. [62]	Ref. [34]
Median Filter (2, 2)	**0.99712**	0.9920	0.989
Median Filter (3, 3)	**0.99710**	0.9890	0.982
Median Filter (5, 5)	**0.99702**	0.9710	0.9500
Median Filter (7, 7)	**0.99681**	0.9510	0.9140
Median Filter (9, 9)	**0.99652**	0.9330	0.8820

Table 13 JPEG compression attacks.

Attack	NC		
	Proposed Scheme	Ref. [62]	Ref. [34]
JPEG compression Q = 5	**0.99686**	0.9520	0.9860
JPEG compression Q = 10	**0.99695**	0.9720	0.9950
JPEG compression Q = 20	**0.99697**	0.9830	0.9960
JPEG compression Q = 25	0.99699	0.9850	**0.9980**
JPEG compression Q = 30	0.99701	0.9870	**0.9980**
JPEG compression Q = 40	0.99704	0.9880	**0.9990**
JPEG compression Q = 50	0.99706	0.9900	**0.9990**
JPEG compression Q = 70	0.99708	0.9940	**0.9990**

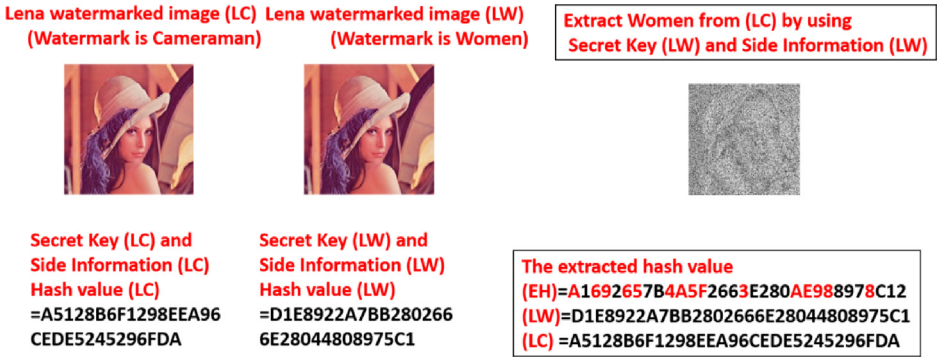

Fig. 11 Resistance against FPP scenario 1.

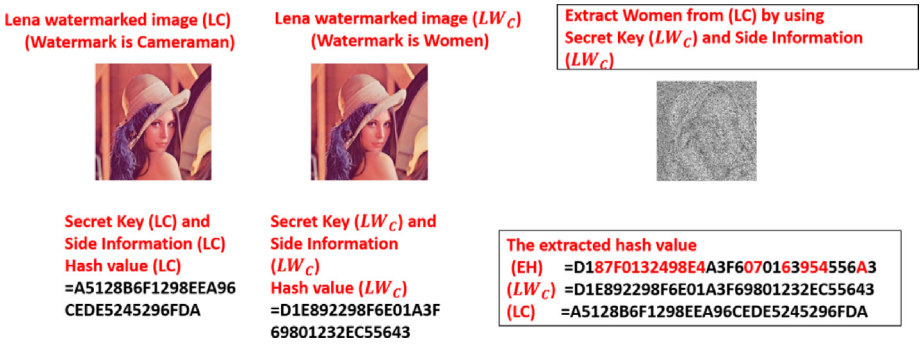

Fig. 12 Resistance against FPP scenario 2.

side information for each watermarking instance are different. Thus, extracting W from LC using the secret key and side information associated with LW (and vice versa) will result in a mismatch of hash values. The two embedding processes have different secret keys and these keys generate different chaotic points for α and β values. α and β matrices of the two embedding processes are different and the two bit-planes are also different. This results in two completely different hash values.In this scenario, an adversary can extract a watermark by using his or her side information and secret key. However, the hash value of the originally embedded watermark is kept by the TTP along with the host image. Anyone claiming ownership of the host image must provide the corresponding hash value to verify the correctness of the watermark. As the hash values do not match, this FPP scenario is circumvented as shown in Fig. 11.

- **FPP Scenario 2:** Let the host image and its owner's watermark be denoted as L and C respectively. C is embedded into L to obtain the watermarked image, secret key, and side information. An adversary embeds another watermark, W into LC, resulting in a new watermarked image, LC_W, another secret key, and a new set of side information. The adversary then attempts to extract W from LC using the secret key and side information associated with LC_W. The adversary's hash value will not match the hash value that corresponds to LC because the secret keys are different.

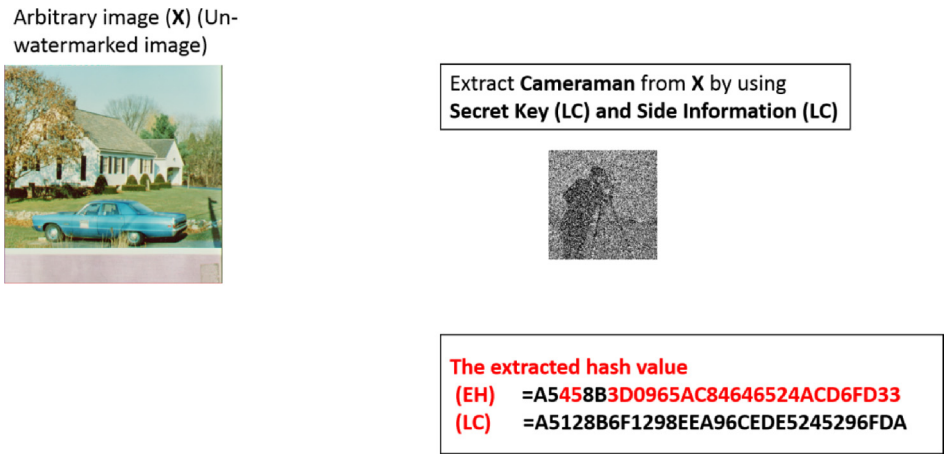

Fig. 13 Resistance against FPP scenario 3.

Thus, the illegal ownership claim is thwarted. The sensitivity of the hash values is due to the sensitivity of the chaotic maps used in the proposed scheme.

- **FPP Scenario 3:** Let the host image and its owner's watermark be denoted as L and C respectively. C is embedded into L to produce the watermarked image, LC, the secret key, and side information. In this scenario, an adversary uses any randomly selected host image X and tries to extract C from X by using the side information of LC. If this succeeds, the adversary can claim ownership of X without having to embed any watermark into X. In this scenario, there are two sub-cases. Firstly, if the host image X is not registered with the TTP, then there is no hash value. Thus, the TTP will stop the verification process because the host image is not included in its directory. In the second sub-case, the host image X is included in the database of the TTP. Each host image has a unique hash value. Wrongful ownership claims can then be detected when hash values are being compared. Therefore, an adversary cannot claim ownership for an arbitrary image by using his or her own secret key and side information.

5.3. Secret key sensitivity

For security and resistance against FPP, the proposed scheme must be sensitive to secret key changes. Hash values for the watermarked image and extracted image should differ even if only a single bit of the secret key has been flipped. In addition, by using an incorrect secret key, an adversary cannot obtain the originally embedded watermark. We analyze the proposed scheme's secret key sensitivity, by observing the results of using keys that differ in just one bit to embed the same watermark into the same host image. Let K_W be the secret key is used to embed a watermark W into a host color image H to produce a watermarked image H_W. To obtain a forged key, $\hat{K_W}$ we flip one bit of K_W . We use K_W and $\hat{K_W}$ to extract W from H_W. We then calculate the NC values for the resulting watermarks. The experiment is repeated 128 times by flipping each of the 128 bits of K_W. Fig. 14 and Fig. 15 show the resulting NC values and the extracted watermark when one secret key bit has been flipped. We can clearly see that even a 1-bit change will lead to noisy images with low NC values.

We now observe how a 1-bit change to the secret key affects the final hash value. The logistic-G and sine-G maps are hypersensitive to small changes to their inputs as depicted in Fig. 4. These changes will lead to a new set of chaotic points being generated, affecting various parts of the scheme such as the sorting of the bit-planes and sub-bands of the host color images. Furthermore, the encryption of the bit-planes depends on the chaotic points. Thus, a different watermark image will be extracted if the secret key is changed. We toggle a single bit of the secret key in different positions and generate the corresponding hash values of the watermarked images. The hash values are represented as hexadecimal characters for visual comparison purposes. We denote the hash values as HV_i and their corresponding secret keys as K_i, where i is the number of changes in the secret key that has been performed. Fig. 16 shows the differences between the resulting hash values for different keys. The change of even a single secret key bit results in a totally different hash value.

5.4. Flexibility

The flexibility of a watermarking scheme refers to its ability to embed various sizes of watermark images or the possibility of embedding them into different sub-bands. In this section, the flexibility of the proposed scheme is studied in terms of watermark image size. Host and watermark images with varying dimensions were used. The watermark was embedded into eight sub-bands to maximize capacity. The proposed scheme cannot embed watermarks with the same dimension as the host image because that would require more than 12 sub-bands. Thus, the maximum capacity is $\frac{M}{2} \times \frac{N}{2}$. To accommodate smaller watermarks that are not equal to the size of the sub-bands, we duplicate pixels from rows and columns so that they are equally divided into subsections that have the same dimension as the sub-bands. The watermark division process is summarized in Algorithm 2.

Algorithm 2. Watermark Division

Data: Input watermark W, where M_W and N_W are height and width. Inp image H, where M_S and N_S are height and width of sub-band
Result: W_k, where k denotes number of subsections

```
if M_W == M_S and N_W == N_S then
    Embed the watermark W into sub_bands of H;
else
    if M_W == M_S and N_W < N_S then
        Duplicate the last pixel columns (N_S − N_W);
        Embed the watermark W into sub − bands H;
    else
        if M_W < M_S and N_W == N_S then
            Duplicate the last pixel rows (M_S − M_W);
            Embed the watermark W into sub − bands H;
        else
            Duplicate the last pixel rows (M_S − M_W);
            Duplicate the last pixel columns (N_S − N_W);
            Embed the watermark W into sub − bands H;
```

To numerically analyze the flexibility of the proposed scheme, the PSNR and NC values for the different sizes of the host and watermark images are calculated. The NC values are calculated after removing any duplicated pixels. Results are tabulated in Table 14. The results indicate that the proposed scheme can successfully embed watermark images with various dimensions into the eight sub-bands of the host images. In addition, the proposed scheme has high imperceptibility and high robustness for all of these sizes. The proposed scheme, therefore, has the flexibility to handle various image sizes as well as the potential to be embedded in sub-bands with high imperceptibility.

5.5. Comparative analysis

In general, SVD-based watermarking schemes rely on two general strategies to embed the watermark information into a host object. SVD is applied and three matrices are generated U, V, and S, which are frequency coefficients. In the first strategy, the owner embeds the watermark information directly into the singular values of S under the control of scaling factors to manage the embedding level. Direct embedding is susceptible to FPP attacks since singular values only contain luminance values, which has a minimal effect on the overall structure of the host image. Singular values have a low impact on the digital image and will not induce distortions during the extraction process after the SVD transformation is reversed. In the second strategy, the watermark information is decomposed by SVD and the singular values S_W are embedded into the singular values S_H of the host image under control of the scaling factors. The singular vectors contain the geometrical and structural information of the digital image and are always kept as side information for watermark extraction. The second strategy suffers from FPP issues. Although the proposed scheme is based on the first strategy and the chaotic MSF values control the embedding strategy, it overcomes FPP through the use of chaotic maps, hash values, encrypted bit-planes, and the secret key, achieving both high imperceptibility and robustness.

SVD-based image watermarking schemes can achieve high embedding capacity [70]. However, schemes that rely on embedding singular vectors or principal components are susceptible to image manipulation and geometric attacks [39,40,15]. As the singular vectors mainly contain structural information about the watermark, small changes to the left or right of the singular vectors will have an adverse effect on the extracted watermark.

In order to make a generic comparison, the proposed scheme is compared to other existing schemes that used SVD transformation in their designs. Table 15 compares the proposed scheme with these schemes based on a set of generic performance factors. The proposed scheme is able to avoid FPP attacks and achieve a desirable trade-off between robustness and imperceptibility due to its embedding strategy and random MSF. It can also support various watermark sizes and sub-bands, unlike the other schemes that focus on just one or two sub-bands. MSF parameters are generated randomly, leading to a good trade-off between watermarking requirements. Overall, the proposed scheme achieves a successful trade-off between watermarking requirements (security, capacity, robustness, and imperceptibility) as compared to its peers. Table 16 shows the comparison of the average NC values with other existing schemes, indicating that the proposed scheme performs at least on par or better than the other schemes. These desirable properties are due to the proposed scheme's new embedding strategy that embeds the bit-planes into the singular values under the control chaotic maps.

The proposed scheme uses different sub-bands from color host images and bit-planes from gray images under the control of a secret key and chaotic maps. These are the defining features of the proposed watermarking scheme that differentiates it from prior schemes. The computational complexity of the proposed scheme depends on the different factors such as:

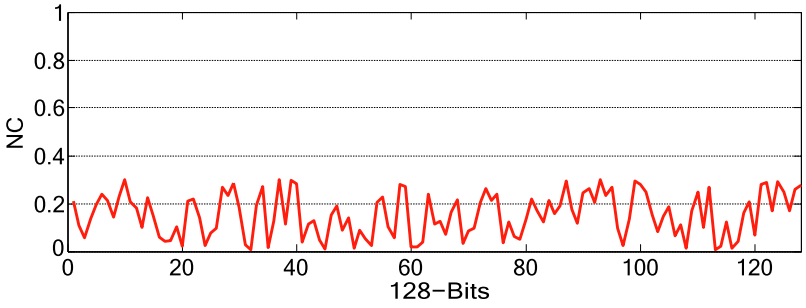

Fig. 14 Secret key sensitivity of the proposed scheme.

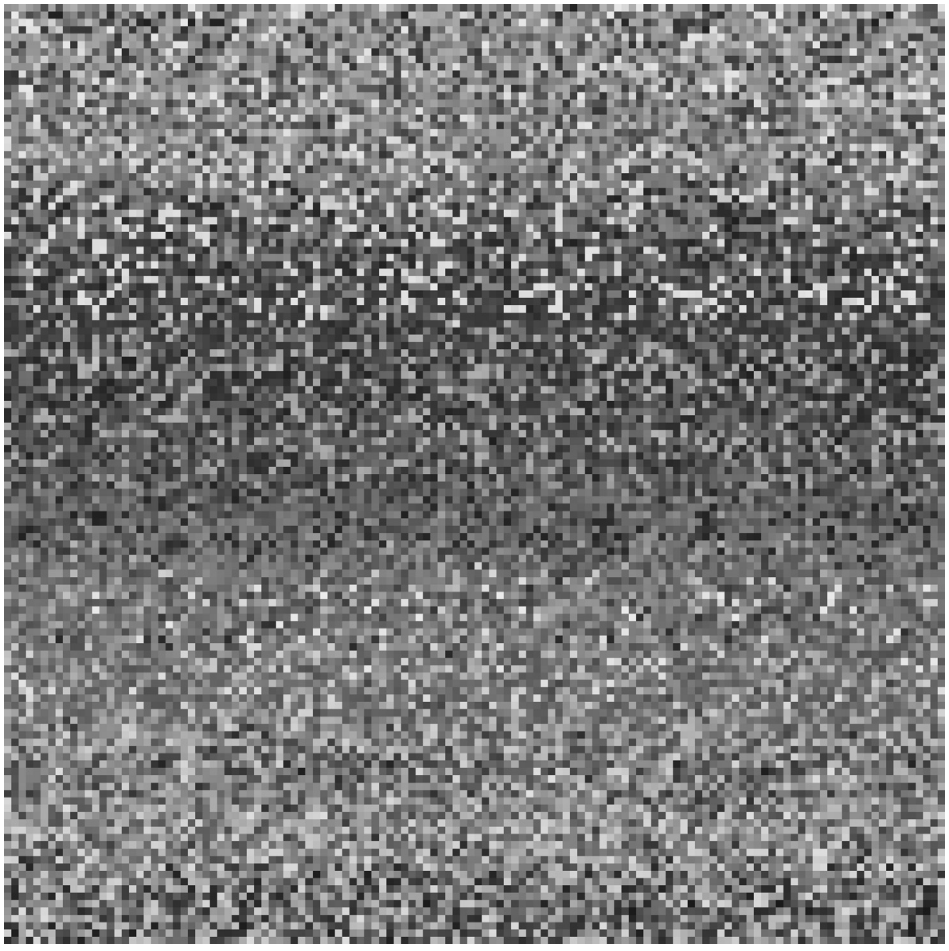

Fig. 15 Extracted noise watermark due to incorrect secret key.

K_0=00112233445566778899AABBCCDDEEFF
K_1=10112233445566778899AABBCCDDEEFF
K_2=00112233445566778799AABBCCDDEEFF
K_3=00112233445566778899AABBCCDDEEFE
K_4=01112233445566778899AABBCCDDEEFF

HV_0=F565F56ACAA1036A2FE264AC639010100
HV_1=9241011236FA0102CC5E5E125C22E41AB
HV_2=851C434504AA44EB10524BBA44F4FE4B
HV_3=C101AB144FF484104E00EA4414514CC89
HV_4=F4FE488EA651010547154FBBC41A5EF09

Fig. 16 Hash bits after changing 1 bit of the secret key (four cases).

Table 14 PSNR and NC values of different sizes of the host images and watermark images.

Host Image	Watermark Image			
128 × 128	32 × 32	65 × 32	32 × 65	65 × 65
	PSNR = 52.33	PSNR = 50.45	PSNR = 49.31	PSNR = 47.74
	NC = 0.99411	NC = 0.99221	NC = 0.99512	NC = 0.99391
320 × 320	32 × 32	90 × 60	72 × 90	120 × 120
	PSNR = 53.51	PSNR = 51.28	PSNR = 53.12	PSNR = 53.22
	NC = 0.99732	NC = 0.99742	NC = 0.99735	NC = 0.99742

Table 15 Comparative analysis of existing SVD-based watermark schemes.

Reference	Embedding Sub-Bands	Type of Transforms	Watermark Size	Scaling Factor	Solve FPP	Embedding Procedure
Proposed Scheme	8 random sub-bands	IWT + SVD	$\frac{M}{2} \times \frac{N}{2}$ size or smaller than	MSF (Chaotic)	Yes	A watermark image is decomposed into bit-planes. Each plane is encrypted and embedded into the eight singular values of the sub-bands.
Ref.[30]	DC	DCT + SVD	64×64	MSF(DE)	Yes	Divide the host image into non-overlapping blocks and DCT is applied on each block, the DC values are selected and SVD is applied. The watermark is embedded in S values.
Ref.[13]	HL and LH	DWT + SVD	256×256 and 12 characters	SSF	No	Divide the watermark image into equal halves and embed into the singular values of two sub-bands, another text watermark is also embedded into the host image.
Ref.[71]	LL_3 and HH_3	DCT + DWT + SVD	128×128	MSF (PSO)	Yes	Encrypt the watermark and embed into the the singular values of the LL_3 and HH_3 sub-bands of the host image.
Ref.[19]	All	RT + DWT + SVD	33×33	SSF	No	S_W was embedded into S_H
Ref.[42]	ALL	DWT + SVD	256×256	MSF (PSO)	Yes	PCs of the watermark were embedded into the host image
Ref.[72]	LH_3	DWT + SVD	32×32	MSF (MOACO)	Yes	S_W was embedded into S_H of the host image, U_W and V_W were hashed and stored as private key
[68]	LL_2	DWT + DCT + SVD	256×256	SSF	No	Apply second level DWT on the host image and then apply DCT to the LL sub-band. SVD is applied and S is embedded into the S of the host image.
Ref.[38]	HH	DWT + SVD	256×256	-	Yes	S_W was replaced by S_H while U_W and V_W are authenticated before extraction
Ref.[20]	LL_3 and HH_3	DWT + SVD + HVS	64×64	(MSF) DE	Yes	The gray-scale watermark was embedded into S_H of LL_3 and HH_3. A secret key was generated by XOR-ing two binary images extracted from LL_3 and watermark
Ref.[73]	All	DWT + SVD	256×256	(MSF) SDE	Yes	Extract PCs from the sub-bands of a watermark after applying DWT and SVD to each sub-band, and alter S_H of host image after applying DWT and SVD
Ref.[70]	All	RSWT + SVD	512×512	SSF	No	Modify S_H of all sub-bands by S_W of watermark image

Table 16 Comparison of average NC among the proposed scheme and four related schemes.

Attack	Average NC				
	Proposed Scheme	Ref. [58]	Ref. [62]	Ref. [34]	Ref. [74]
Salat and Pepper	**0.997**	0.951	0.632	0.995	0.992
Gaussian Noise	**0.997**	0.974	0.638	0.619	0.986
Median Filter	**0.997**	0.988	0.989	0.989	0.994
Cropping	**0.997**	0.791	-	-	0.991
JPEG Compression	**0.997**	0.994	0.952	0.998	0.993

- Decomposition of the watermark image.
- Key generation and chaotic parameter generation.
- Encryption of the bit-planes.
- IWT decomposition of the host image.
- SVD Decomposition of the host image.
- Embedding process.
- Hash generation.

These steps contribute towards the computational complexity of the proposed scheme. However, the proposed scheme still outperforms other schemes that rely on optimization algorithms to identify optimal MSF parameters. The proposed scheme requires only 3.125s for embedding and 3.045s for extraction for a host image with a dimension of 512×512 and a watermark with a dimension of 256×256.

6. Conclusion

A new SVD-based color image watermarking scheme based on the frequency domain and bit-plane decomposition was proposed. The secret key was used to generate a chaotic sequence

that is used for encryption, permutation (reordering), and generating MSF parameters. The host color image was transformed into 12 sub-bands and then reordered based on the chaotic sequence before applying SVD to the first eight sub-bands. The grayscale watermark image was decomposed into eight bit-planes, each bit-plane was encrypted using the chaotic sequence and also reordered. Each bit-plane was embedded into the singular values of the corresponding sub-band of the host color image. The MSF parameters were generated based on the chaotic sequence and leads to a desirable trade-off between robustness and imperceptibility. MSF parameters were updated in each embedding process by comparing between previously generated MSF parameters and S_i^{New} values for improved security, robustness, and imperceptibility. At the end embedding processes, a hash value was generated, unique to the host and watermark images. This hash value was used to overcome the FPP issues. During the extraction process, a hash value was generated and compared to the stored hash value. A match implies that the extracted watermark was indeed the original watermark. Otherwise, a hash mismatch indicates the presence of a fake or forged watermark. Thus, the proposed scheme can overcome FPP issues that many other schemes are susceptible to. The proposed scheme also achieved the desired trade-off between robustness and imperceptibility, and high sensitivity to small changes in the secret key. The proposed scheme extracted watermark images with high NC scores under various geometrical and non-geometrical attacks. In future work, the proposed scheme can be improved in terms of capacity and the ability to support color watermark images.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Kumar, A.K. Singh, P. Kumar, A recent survey on image watermarking techniques and its application in e-governance, Multimedia Tools Appl. 77 (3) (2018) 3597–3622.

[2] J.-S. Pan, H.-C. Huang, L.C. Jain, Intelligent Watermarking Techniques, vol. 7, World scientific, 2004.

[3] H.-C. Huang, Information Hiding and Applications, vol. 227, Springer, 2009.

[4] S.N. Bal, M.R. Nayak, S.K. Sarkar, On the implementation of a secured watermarking mechanism based on cryptography and bit pairs matching, J. King Saud Univ.-Comput. Informat. Sci. (2018).

[5] F. Ernawan, M.N. Kabir, An improved watermarking technique for copyright protection based on tchebichef moments, IEEE Access 7 (2019) 151985–152003.

[6] L.-Y. Hsu, H.-T. Hu, A reinforced blind color image watermarking scheme based on schur decomposition, IEEE Access 7 (2019) 107438–107452.

[7] C. Saha, M. Hossain, M. Rahman, et al, Nsct-based robust image watermarking in dc components of apdcbt using singular value decomposition, Iran J. Comput. Sci. 4 (3) (2021) 133–145.

[8] S. Singh, V.S. Rathore, R. Singh, Hybrid nsct domain multiple watermarking for medical images, Multimedia Tools Appl. 76 (3) (2017) 3557–3575.

[9] S. Singh, V.S. Rathore, R. Singh, M.K. Singh, Hybrid semi-blind image watermarking in redundant wavelet domain, Multimedia Tools Appl. 76 (18) (2017) 19113–19137.

[10] J.C. Patra, J.E. Phua, C. Bornand, A novel dct domain crt-based watermarking scheme for image authentication surviving jpeg compression, Digital Signal Process. 20 (6) (2010) 1597–1611.

[11] Premaratne, Ko, A novel watermark embedding and detection scheme for images in dft domain, in: Image Processing And Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465), vol. 2, 1999, pp. 780–783, doi:10.1049/cp:19990430.

[12] E. Najafi, A robust embedding and blind extraction of image watermarking based on discrete wavelet transform, Mathe. Sci. 11 (4) (2017) 307–318.

[13] A. Anand, A.K. Singh, An improved dwt-svd domain watermarking for medical information security, Comput. Commun. 152 (2020) 72–80.

[14] N.A. Loan, N.N. Hurrah, S.A. Parah, J.W. Lee, J.A. Sheikh, G. M. Bhat, Secure and robust digital image watermarking using coefficient differencing and chaotic encryption, IEEE Access 6 (2018) 19876–19897.

[15] N.M. Makbol, B.E. Khoo, T.H. Rassem, K. Loukhaoukha, A new reliable optimized image watermarking scheme based on the integer wavelet transform and singular value decomposition for copyright protection, Inf. Sci. 417 (2017) 381–400.

[16] S. Maheshkar et al, Region-based hybrid medical image watermarking for secure telemedicine applications, Multimedia Tools Appl. 76 (3) (2017) 3617–3647.

[17] N.R. Zhou, A.W. Luo, W.P. Zou, Secure and robust watermark scheme based on multiple transforms and particle swarm optimization algorithm, Multimedia Tools Appl. 78 (2) (2019) 2507–2523.

[18] A.K. Singh, B. Kumar, S.K. Singh, S. Ghrera, A. Mohan, Multiple watermarking technique for securing online social network contents using back propagation neural network, future generation computer systems 86 (2018) 926–939.

[19] S. Rastegar, F. Namazi, K. Yaghmaie, A. Aliabadian, Hybrid watermarking algorithm based on singular value decomposition and radon transform, AEU-Int. J. Electron. Commun. 65 (7) (2011) 658–663.

[20] M. Ali, C.W. Ahn, P. Siarry, Differential evolution algorithm for the selection of optimal scaling factors in image watermarking, Eng. Appl. Artif. Intell. 31 (2014) 15–26.

[21] V. Aslantas, An optimal robust digital image watermarking based on svd using differential evolution algorithm, Opt. Commun. 282 (5) (2009) 769–777.

[22] C. Kumar, A.K. Singh, P. Kumar, Dual watermarking: An approach for securing digital documents, Multimedia Tools Appl. (2019) 1–16.

[23] A.K. Abdulrahman, S. Ozturk, A novel hybrid dct and dwt based robust watermarking algorithm for color images, Multimedia Tools Appl. 78 (12) (2019) 17027–17049.

[24] M. Ali, C.W. Ahn, M. Pant, An efficient lossless robust watermarking scheme by integrating redistributed invariant wavelet and fractional fourier transforms, Multimedia Tools Appl. 77 (10) (2018) 11751–11773.

[25] X.-B. Kang, F. Zhao, G.-F. Lin, Y.-J. Chen, A novel hybrid of dct and svd in dwt domain for robust and invisible blind image watermarking with optimal embedding strength, Multimedia Tools Appl. 77 (11) (2018) 13197–13224.

[26] E. Najafi, K. Loukhaoukha, Hybrid secure and robust image watermarking scheme based on svd and sharp frequency localized contourlet transform, J. Informat. Security Appl. 44 (2019) 144–156.

[27] M. Ali, C.W. Ahn, M. Pant, P. Siarry, An image watermarking scheme in wavelet domain with optimized compensation of singular value decomposition via artificial bee colony, Inf. Sci. 301 (2015) 44–60.

[28] N.M. Makbol, B.E. Khoo, T.H. Rassem, Block-based discrete wavelet transform-singular value decomposition image watermarking scheme using human visual system characteristics, IET Image Process. 10 (1) (2016) 34–52.

[29] S. Roy, A.K. Pal, A robust blind hybrid image watermarking scheme in rdwt-dct domain using arnold scrambling, Multimedia Tools Appl. 76 (3) (2017) 3577–3616.
[30] M. Ali, C.W. Ahn, M. Pant, A robust image watermarking technique using svd and differential evolution in dct domain, Optik-I. J. Light Electron Opt. 125 (1) (2014) 428–434.
[31] H.-C. Ling, R.C.-W. Phan, S.-H. Heng, On the security of a hybrid svd-dct watermarking method based on lpsnr, in: Pacific-Rim Symposium on Image and Video Technology, Springer, 2011, pp. 257–266.
[32] D. Singh, S.K. Singh, Dwt-svd and dct based robust and blind watermarking scheme for copyright protection, Multimedia Tools Appl. 76 (11) (2017) 13001–13024.
[33] A.K. Singh, Improved hybrid algorithm for robust and imperceptible multiple watermarking using digital images, Multimedia Tools Appl. 76 (6) (2017) 8881–8900.
[34] N.M. Makbol, B.E. Khoo, A new robust and secure digital image watermarking scheme based on the integer wavelet transform and singular value decomposition, Digital Signal Process. 33 (2014) 134–147.
[35] I.J. Cox, M.L. Miller, A.L. McKellips, Watermarking as communications with side information, Proc. IEEE 87 (7) (1999) 1127–1141.
[36] K. Loukhaoukha, J.-Y. Chouinard, M.H. Taieb, Optimal image watermarking algorithm based on lwt-svd via multi-objective ant colony optimization, J. Informat. Hiding nd Multimedia Signal Process. 2 (4) (2011) 303–319.
[37] K. Loukhaoukha, J.-Y. Chouinard, Hybrid watermarking algorithm based on svd and lifting wavelet transform for ownership verification, in: 2009 11th Canadian Workshop on Information Theory, IEEE, 2009, pp. 177–182.
[38] A.K. Gupta, M.S. Raval, A robust and secure watermarking scheme based on singular values replacement, Sadhana 37 (4) (2012) 425–440.
[39] C. Jain, S. Arora, P.K. Panigrahi, A reliable svd based watermarking schem, arXiv preprint arXiv:0808.0309.
[40] R. Liu, T. Tan, An svd-based watermarking scheme for protecting rightful ownership, IEEE Trans. Multimedia 4 (1) (2002) 121–128.
[41] S. Roy, A.K. Pal, An svd based location specific robust color image watermarking scheme using rdwt and arnold scrambling, Wireless Pers. Commun. 98 (2) (2018) 2223–2250.
[42] R.-S. Run, S.-J. Horng, J.-L. Lai, T.-W. Kao, R.-J. Chen, An improved svd-based watermarking technique for copyright protection, Expert Syst. Appl. 39 (1) (2012) 673–689.
[43] A. Zear, A.K. Singh, P. Kumar, A proposed secure multiple watermarking technique based on dwt, dct and svd for application in medicine, Multimedia Tools Appl. 77 (4) (2018) 4863–4882.
[44] K. Sreenivas, V.K. Prasad, Fragile watermarking schemes for image authentication: a survey, Int. J. Machine Learn. Cybernet. 9 (7) (2018) 1193–1218.
[45] S. Thakur, A.K. Singh, S.P. Ghrera, A. Mohan, Chaotic based secure watermarking approach for medical images, Multimedia Tools Appl. 79 (7) (2020) 4263–4276.
[46] J. Liu, J. Li, J. Ma, N. Sadiq, U.A. Bhatti, Y. Ai, A robust multi-watermarking algorithm for medical images based on dtcwt-dct and henon map, Appl. Sci. 9 (4) (2019) 700.
[47] M. Alawida, A. Samsudin, J.S. Teh, et al, Digital cosine chaotic map for cryptographic applications, IEEE Access 7 (2019) 150609–150622.
[48] M. Alawida, A. Samsudin, J.S. Teh, Enhanced digital chaotic maps based on bit reversal with applications in random bit generators, Inf. Sci. 512 (2020) 1155–1169.
[49] J. Wang, H. Peng, P. Shi, An optimal image watermarking approach based on a multi-objective genetic algorithm, Inf. Sci. 181 (24) (2011) 5501–5514.
[50] I.A. Ansari, M. Pant, C.W. Ahn, Robust and false positive free watermarking in iwt domain using svd and abc, Eng. Appl. Artif. Intell. 49 (2016) 114–125.
[51] I.A. Ansari, M. Pant, Multipurpose image watermarking in the domain of dwt based on svd and abc, Pattern Recogn. Lett. 94 (2017) 228–236.
[52] M. Moosazadeh, G. Ekbatanifard, A new dct-based robust image watermarking method using teaching-learning-based optimization, J. Informat. Security Appl. 47 (2019) 28–38.
[53] W.H. Alshoura, Z. Zainol, J.S. Teh, M. Alawida, A. Alabdulatif, Hybrid svd-based image watermarking schemes: A review, IEEE Access 9 (2021) 32931–32968, https://doi.org/10.1109/ACCESS.2021.3060861.
[54] M. Alawida, A. Samsudin, J.S. Teh, R.S. Alkhawaldeh, A new hybrid digital chaotic system with applications in image encryption, Signal Process. 160 (2019) 45–58.
[55] M. Alawida, J.S. Teh, A. Samsudin, et al, An image encryption scheme based on hybridizing digital chaos and finite state machine, Signal Process. (2019).
[56] M. Alawida, A. Samsudin, J.S. Teh, Enhancing unimodal digital chaotic maps through hybridisation, Nonlinear Dyn. 96 (1) (2019) 601–613.
[57] L. Chen, J. Chen, G. Zhao, S. Wang, Cryptanalysis and improvement of a chaos-based watermarking scheme, IEEE Access 7 (2019) 97549–97565.
[58] X. Wu, W. Sun, Robust copyright protection scheme for digital images using overlapping dct and svd, Appl. Soft Comput. 13 (2) (2013) 1170–1182.
[59] S. Rawat, B. Raman, A blind watermarking algorithm based on fractional fourier transform and visual cryptography, Signal Process. 92 (6) (2012) 1480–1491.
[60] A. Rani, B. Raman, An image copyright protection scheme by encrypting secret data with the host image, Multimedia Tools Appli. 75 (2) (2016) 1027–1042.
[61] T.-Y. Fan, H.-C. Chao, B.-C. Chieu, Medical image watermarking based on visual secret sharing and cellular automata transform for copyright protection., KSII Trans. Internet Informat. Syst. 12(12) (2018).
[62] N.M. Makbol, B.E. Khoo, Robust blind image watermarking scheme based on redundant discrete wavelet transform and singular value decomposition, AEU-Int. J. Electron. Commun. 67 (2) (2013) 102–112.
[63] N. Zermi, A. Khaldi, R. Kafi, F. Kahlessenane, S. Euschi, A dwt-svd based robust digital watermarking for medical image security, Forensic Sci. Int. 320 (2021) 110691.
[64] N. Zermi, A. Khaldi, M.R. Kafi, F. Kahlessenane, S. Euschi, Robust svd-based schemes for medical image watermarking, Microprocess. Microsyst. 84 (2021) 104134.
[65] C.-C. Lai, A digital watermarking scheme based on singular value decomposition and tiny genetic algorithm, Digital Signal Process. 21 (4) (2011) 522–527.
[66] M.I. Khan, M. Rahman, M. Sarker, I. Hasan, et al., Digital watermarking for image authenticationbased on combined dct, dwt and svd transformation, arXiv preprint arXiv:1307.6328.
[67] N. Harish, B. Kumar, A. Kusagur, Hybrid robust watermarking techniques based on dwt, dct, and svd, Int. J. Adv. Electrical Electron. Eng. 2 (5) (2013) 137–143.
[68] S. Thakur, A.K. Singh, S.P. Ghrera, M. Elhoseny, Multi-layer security of medical data through watermarking and chaotic encryption for tele-health applications, Multimedia Tools Appl. 78 (3) (2019) 3457–3470.
[69] A.K. Singh, M. Dave, A. Mohan, Hybrid technique for robust and imperceptible image watermarking in dwt–dct–svd domain, National Acad. Sci. Lett. 37 (4) (2014) 351–358.
[70] S. Lagzian, M. Soryani, M. Fathy, Robust watermarking scheme based on rdwt-svd: Embedding data in all subbands,

in: 2011 International Symposium on Artificial Intelligence and Signal Processing (AISP), IEEE, 2011, pp. 48–52.

[71] L. Zhang, D. Wei, Dual dct-dwt-svd digital watermarking algorithm based on particle swarm optimization, Multimedia Tools Appl. 78 (19) (2019) 28003–28023.

[72] K. Loukhaoukha, Image watermarking algorithm based on multiobjective ant colony optimization and singular value decomposition in wavelet domain, J. Optim. (2013).

[73] M. Ali, C.W. Ahn, An optimized watermarking technique based on self-adaptive de in dwt–svd transform domain, Signal Process. 94 (2014) 545–556.

[74] W.H. Alshoura, Z. Zainol, J.S. Teh, M. Alawida, A new chaotic image watermarking scheme based on svd and iwt, IEEE Access 8 (2020) 43391–43406, https://doi.org/10.1109/ACCESS.2020.2978186.