International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 5s, 2025
https://www.theaspd.com/ijes.php

Logistic-Map Based Fragile Watermarking Scheme With Dual-Security For Image Integrity Verification In Cloud Computing Environment

Jyoti Rani[1], Rajender Nath[2], and Meena Kumari[3]

[1,2,3]Department of Computer Science and Applications, Kurukshetra University Kurukshetra,
[1]jyoti41nagwan@kuk.ac.in, [2]rnath@kuk.ac.in , [3]sangermeena@gmail.com

Abstract: *With the widespread embrace of cloud computing technology, the utilization of cloud storage has become highly prevalent, given the limited storage capacity of user's local devices. Nevertheless, throughout the processes of transmitting, storing, and downloading digital images to or from the cloud, there is the potential for integrity violations or tampering with the images. Addressing this concern, the paper introduces a fragile dual-security watermarking scheme based on blocks and logistic maps. The purpose of this scheme is to protect digital images authenticity and integrity while they are being stored and transmitted between the user and the cloud. Before watermarking, the image undergoes a jumbling process using a logistic map, providing dual level of security. The proposed scheme employs a combination of arithmetic mean, standard deviation, and logistic map for watermark generation, as well as integrity checking or tamper detection. This results in a notable reduction in execution time, as demonstrated through experimentation when compared to existing schemes. Furthermore, the suggested method efficiently locates and identifies tampering in the event of multiple attack types, such as copy-paste, copy-move, and crop attacks. Moreover, the proposed scheme exhibits excellent imperceptibility, with average PSNR and SSIM values of 51.14178 and 0.9967, respectively, in comparison to existing schemes.*

Keywords: *Cloud computing, integrity verification, tamper detection, tamper localization, fragile, watermarking, spatial-domain watermarking, logistic-map, jumbling, standard deviation, arithmetic mean.*

1. introduction

With the evolution of cloud computing technology and the internet, the exchange of data among users is experiencing significant growth. Due to the limited capacity of users' local devices, the utilization of cloud computing services becomes imperative for accessing unlimited computing and storage resources. Users can seamlessly outsource their data to the cloud for storage and retrieve it when needed, encompassing various data formats such as text, images, audio, and video. This paper specifically focuses on image data, highlighting associated security concerns.

During the upload and download processes from the cloud, there exists a vulnerability wherein attackers can manipulate images using readily available image processing software or techniques. This tampering can go unnoticed, as the altered image may still appear identical to the original. Therefore, there is a pressing need for mechanisms to verify the integrity and authenticity of images both at the cloud server's end during uploading and at the user's end during downloading.

Feasible solutions to address these concerns involve employing hashing, cryptography, and digital image watermarking. By attaching a digital signature to an image, one can effectively identify tampering. However, incorporating a digital signature explicitly introduces challenges, such as the requirement for extra bandwidth or extended transmission time.

Cryptography offers an alternative approach by transforming the plain image into a cipher image using secret keys. Only entities possessing the key can revert the cipher image back to its original form. Nevertheless, this method introduces a challenge where the very essence or impression of the image undergoes a change during the process.

When a digital image is watermarked, the sender creates and embeds a watermark into the original image. The watermark is extracted by the recipient from the received watermarked image, and it is then

recalculated using the same received watermarked image. By comparing them, if the extracted watermark matches the calculated watermark, it means there is no tampering; otherwise, the received image is tampered during transmission. Watermarking emerges as a viable solution for integrity checking during image transmission, as it eliminates the need for explicit signatures (as in digital signatures) since the image itself contains the watermark embedded in it. Furthermore, the impression of the image remains unchanged, unlike in cryptography, making it indistinguishable to the naked eye.

There are two types of watermarks: visible and invisible. Visible watermarks, such as logos or text, serve for owner identification or authentication. Three categories of invisible watermarks exist: robust, semi-fragile, and fragile watermarks. Fragile watermarks are sensitive to any kind of attack or tampering, making them suitable for integrity verification or tamper detection. Robust watermarks can withstand any attack, serving to verify ownership or secure copyright. Semi-fragile watermarks lie between fragile and robust watermarks, being susceptible to some attacks while withstanding others. They are able to recognise areas of damage in an image and extract the undamaged watermark from the unaltered portion.

The two domains in which watermarking can be employed are the spatial and frequency domains. Frequency domain watermarking inserts the watermark into the frequency domain of the image, such as DCT, DWT, or SVD, while spatial domain watermarking involves embedding the watermark into the pixels of the image. Fragile watermarking techniques typically use spatial domain techniques, while frequency domain techniques are employed for robustness. Additionally, block-based watermarking and pixel-based watermarking are both possible. Block-based watermarking begins with the image being split into blocks, and the watermark is generated, embedded, and extracted for each block individually. In pixel-based watermarking, the unit of work for watermark generation, embedding, and extraction is a single pixel.

In order to ensure the integrity and authenticity of digital images while they are being transmitted between users and the cloud, this paper proposes a fragile digital image watermarking scheme based on logistic maps. The scheme provides dual-level security by utilizing an additional level of jumbling.For an additional degree of security, the watermark is inserted into the jumbled image rather than the original one. In order to increase security and cause confusion, logistic maps are used in image jumbling, watermark creation and embedding, and integrity verification. The proposed scheme is block-based, with a carefully chosen block size of 4×4. The complete block is utilized for embedding the watermark, ensuring satisfactory embedding capacity. Careful consideration of block size is crucial, as larger block sizes may not provide precise tamper localization, marking the complete block as tampered if any single pixel is found altered, while smaller block sizes may result in increased time complexity for various watermarking operations.

1.1 Contribution

In particular, this paper's primary contribution can be summed up as follows:

(i) A novel fragile logistic map-based scheme with dual level of security through jumbling is proposed.

(ii) Simple mathematical operations, such as mean and standard deviation is used to calculate the watermark instead of hashing like SHA-1, SHA-256, MD5 etc, which prove to be more efficient in terms of time complexity.

(iii) The detection and localization of tampering in images during transmission between the cloud and users are successfully achieved.

(iv) Operating in the spatial domain, the proposed scheme boasts a lower time complexity than frequency domain schemes.

(v) Calculating the watermark directly from the image eliminates the need for an explicit watermark.

(vi) The watermark is embedded into the entire block rather than just a portion of it, ensuring a satisfactory embedding capacity for the proposed scheme.

(vii) Experiments show that the suggested scheme is resistant to multiple kinds of attacks, such as copy-paste, copy-move, and crop attacks.

(viii) An average PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) demonstrate the good imperceptibility of the proposed scheme.

1.2 Related Work

To address the security of images, researchers have proposed several schemes for image integrity verification or tamper detection using watermarking. These images may be general, medical (such as X-ray, CT, MRI), satellite, and so forth. Different researchers have focused on tampering in various types of images, as discussed in the following lines.

A blind watermarking approach [1] has been proposed for securing medical images, utilizing MD5 hashing and a combination of DWT-SVD for watermark generation and embedding, respectively. Schur decomposition could be an alternative to SVD to reduce computing time. For the integrity and authenticity of medical images, a different reversible fragile watermarking scheme using a 8 ×8 size block has been proposed [2]. Two techniques are used to generate authentication bits: the WBE method, which combines block, exception, and watermark bits, and the MD5 hash algorithm, which proves more accurate in tamper detection according to experiments.

A block-based fragile watermarking scheme [3] employs simple XOR and mean operations for watermark embedding, using a block of size 4×4. This block is further subdivided into two subblocks, embedding different information into each. To address the issue of less embedding capacity resulting from using only half of the block, another pixel-wise embedding scheme [4] is proposed, based on the logistic map for watermark generation and XOR operation for embedding. This scheme, being pixel-based, declares only the tampered pixel as altered, making it suitable for applications in medical imaging and telemedicine. Nevertheless, in events of copy-paste, copy-move, and constant average attacks, the Gull et al. [3] scheme failed to identify image tampering. To overcome this, another simple spatial-domain fragile watermarking scheme [5] is proposed, using SHA-1 hashing for watermark generation and the simple LSB method for embedding. This scheme is capable of detecting tampering in any type of attack, unlike Gull et al. [3]. Another region-based reversible scheme that distinguishes between the ROI (region of interest) and RONI (region of non-interest) was also proposed by Bhalerao et al. [6] for the security of medical images. It can detect tampering in both ROI and RONI but can only recover the image in ROI. In a research paper [7], Aditya Kumar Sahu et al. also proposed two schemes for watermark bit generation that are both based on the logistic map. Although both schemes are responsible for image's tamper detection and localization, one is irreversible and the other is reversible. The schemes can be improved for self-recovery of tampered pixels.

An LSB-based fragile watermarked scheme, utilizing logistic map and Arnold cat map with XOR operation, has been proposed for image tamper detection and authentication [8]. This scheme stands against various types of attacks or noise. However, the scheme employs some static value for watermark generation, potentially compromising image security. Another fragile watermarking scheme [9] for image integrity protection has been proposed, also using the logistic map and modulus operator for watermark generation and embedding, with the intention of identifying and locating image tampered areas. One more scheme [10] scrambled the image using Arnold transformation to increase security, utilizing a hybrid approach of spatial domain and frequency domain watermarking for watermark generation and tamper detection & localization, specifically SHA-256 and DCT. However, this increases the time complexity as the frequency domain is involved.

Some researchers worked on schemes combining biometrics and watermarking for authentication, confidentiality, and integrity control of medical images during transmission [11]. Here, biometrics are used to generate keys for cryptography. Specifically, tamper detection and localization of satellite images during transmission is the focus of the self-embedding fragile scheme [12], which can detect intentional as well as unintentional attacks pixel-by-pixel.

During image transmission to the cloud for storage, integrity and confidentiality could be violated. To address this, a fragile watermarking scheme [13] has been proposed, utilizing two watermarks: one for tamper detection and the other for recovery. It also scrambles the image before watermarking to increase security. However, the scheme's drawback lies in its complexity level in terms of computation.

Problem Formulation: The literature makes clear that a significant number of algorithms [1–14] have been proposed for the security and integrity of images during transmission, but only small subset [4,7–10,13] of them have scrambled the image or used the logistic map to increase the security. Additionally, only few [13,14] of them have specifically focused on the security of images during transmission between the cloud and users. None have used the combination of arithmetic mean and standard deviation for watermark generation and tamper detection. There is a need for a novel scheme with efficient time complexity, tamper detection for all types of attacks, better imperceptibility & embedding capacity, and increased security level. This paper proposes a novel scheme that combines the arithmetic mean, standard deviation, and logistic map for watermark generation and tamper detection & localization, resulting in minimal execution time, better imperceptibility, and embedding capacity. Additionally, image jumbling is performed before watermark embedding to further increase the security level in the proposed scheme.

1.3 Organization

This paper's structure is set up as follows: A thorough description of the proposed scheme, along with its system model and some initial concepts, are provided in Section 2. The experimental outcomes of the proposed scheme are examined in Section 3. The paper is finally concluded in Section 4.

2. proposed scheme

In this section, the fundamental concepts related to the proposed scheme are explored, including the system model and logistic map. Subsequently, the proposed scheme for image integrity verification based on watermarking is discussed.

2.1 System Model of the Proposed Scheme

Fundamentally, the system model of the proposed scheme comprises three entities: the cloud server, the data owner or image uploader, and the users or image downloaders, as illustrated in Figure 1. The responsibilities of these entities can be delineated as follows:

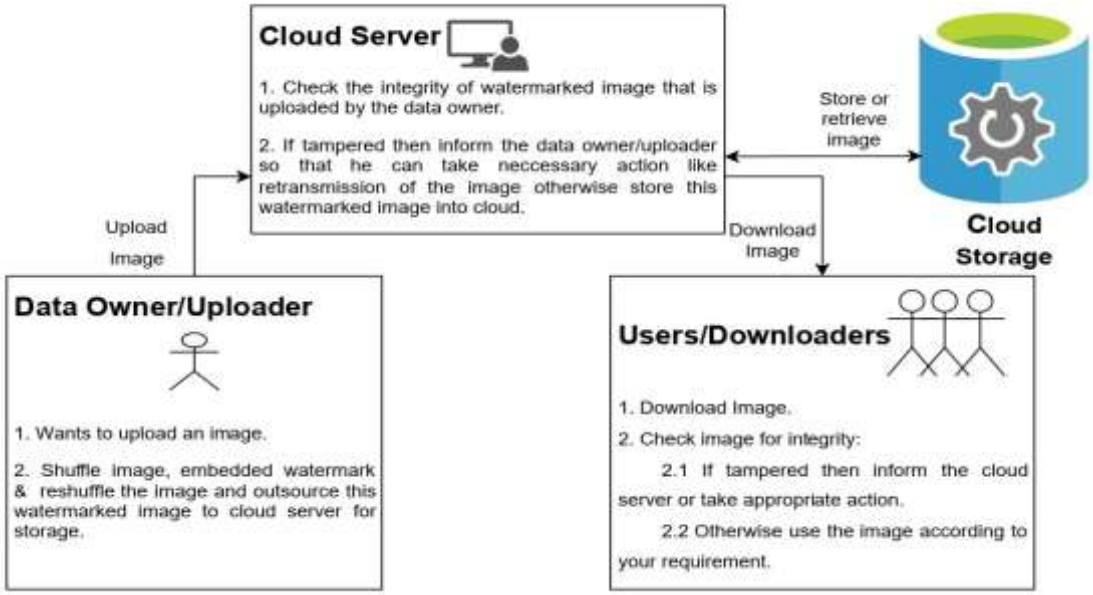

Figure 1: System Model of the Proposed Scheme

Data Owner: The data owner or uploader is an entity looking to outsource its image to cloud storage due to limited local storage capacity. First, the given image is jumbled to increase security. Subsequently, a watermark is created and incorporated into every block to detect any tampering at the downloader's site or the cloud server. Finally, the image undergoes re-shuffling to produce the watermarked image, which is then outsourced to the cloud server for storage.

Cloud Server: The Cloud server has a dual role, offering the data owner unlimited storage capacity while also shouldering the responsibility of detecting tampering in the uploaded images. Given that the image may face tampering from malicious entities or attackers during transmission to the cloud, the Cloud server is instrumental in checking the integrity of the uploaded content. As the proposed scheme employs a fragile watermarking approach, any tampering with the image results in the distortion or destruction of the embedded watermark. This inherent fragility enables effective tamper detection or integrity verification for the uploaded images.

Users: Users or downloaders seek access to images uploaded by the data owner, who may also be a user. A user submits a request to the cloud server to gain access. The user is then given the requested image by the cloud server. Upon receiving the image, its integrity is verified using a tamper detection algorithm. If the image remains unaltered, the user proceeds to use it; however, if tampering is detected, the cloud server is promptly informed so that necessary actions can be taken.

2.2 Logistic Map

A simple polynomial equation called the logistic map is used to create a 1-D random sequence. Its versatility finds application in cryptography, particularly in image security aspects such as image encryption, watermark bit generation, and image shuffling. One notable feature of the logistic map is that, when initialised with the same seed value, it can generate the same sequence at the sender and recipient sides. Consequently, this seed value serves as a secret key between the sender and receiver, providing a basis for secure communication. The following equation defines the logistic map:

$$z_i = n \times z_{i-1} \times (1 - z_{i-1}) \quad (1)$$

Here, the domain of n is $0 < n \leq 4$ and z_0 is $0 \leq z_0 \leq 1$. To achieve optimal chaotic behaviour, the value of n should be chosen within 3.57 and 4 (inclusive) [15]. The Combination of n and z_0 is referred to as the seed value. The values of random sequence generated by logistic map fall within the range of 0 and 1.
In the proposed scheme, the logistic map sequence serves a dual purpose: for image jumbling and watermark bit generation. The length of the chaotic sequence that generated is equal to the number of pixels in the given block. This sequence is then sorted, and the sorted order is employed for jumbling. For instance, if the first value, after sorting, moves to the fourth place, then after jumbling, the first pixel of the block will relocate to the fourth place, and so on. Additionally, another logistic map sequence, with a size equal to the total count of blocks present in the image, is generated for watermark generation.

2.3 Logistic-Map Based Fragile Watermarking Scheme (Proposed)

To provide a dual layer of security and guarantee the authenticity and integrity of images sent over the cloud, a fragile watermarking scheme is presented in this section. To achieve this dual security layer, the image undergoes a jumbling process using a logistic map, and after that, a watermark is added to the jumbled image for authentication or integrity checking. The proposed scheme comprises two primary phases, (i) Watermark Generation and Embedding (ii) Tamper Detection or Integrity Verification. The section that follows provides a detailed explanation of each of these stages.

2.3.1 Watermark Generation and Embedding

The process of generating and embedding the watermark into the given image is illustrated through an activity diagram in Figure 2. Furthermore, the steps are elaborated upon in detail below:

(i) Input the original image.

(ii) Set the original image's least significant bit (LSB) to zero.

(iii) Then, make non-overlapping 4 × 4 blocks out of the image.

(iv) Determine the average and standard deviation for every block, then multiply the results by the respective logistic map sequence value.

(v) Convert the rounded values of resulting values into binary and concatenate them to form16-bits watermark for each block.

(vi) Apply a jumbling process to the original image using the logistic map to enhance the security level.

(vii) Create non-overlapping 4 × 4 blocks out of the resulting jumbled image.

(viii) Embed the 16-bit watermark obtained in step (v) into the respective blocks of the jumbled image using the LSB substitution method.

(ix) Perform the inverse jumbling on the resulting image to generate the watermarked image.

The steps from Step (i) to Step (v) are responsible for generating watermark bits, while the next steps take care of adding this watermark to the original image, producing a watermarked image that is ready

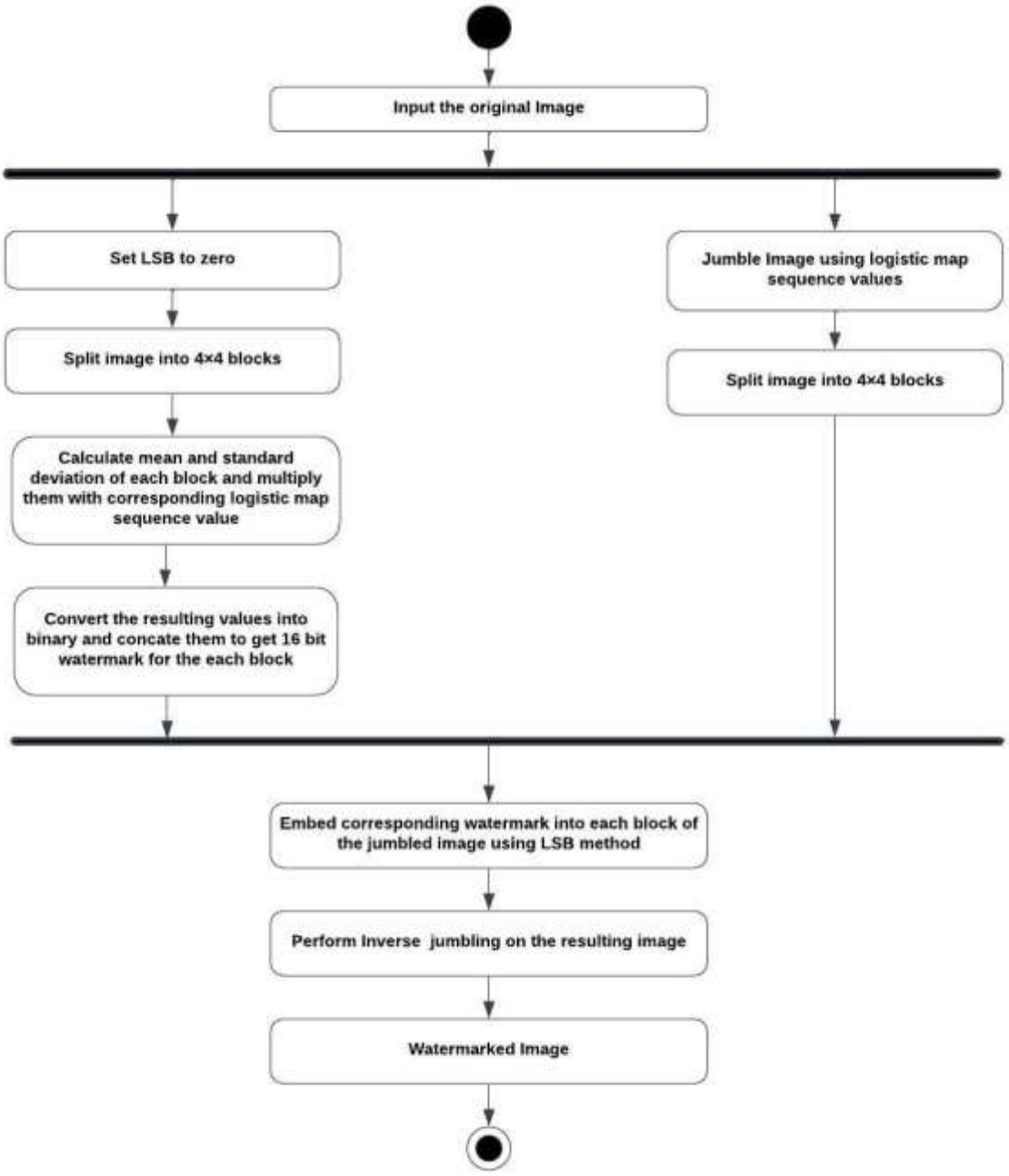

Figure 2: Activity diagram for Watermark Generation and Embedding

to be sent to the cloud. This process of generating watermark bits and embedding is illustrated with a simplified example depicted in Figure 3. In this example, the image is assumed to be a single 4 × 4 block image, providing a concise explanation of the watermarking process. The image is first split into 4 × 4 size blocks, and each block is processed in this manner once. The final watermarked image is created by combining these watermarked blocks.

2.3.2 Tamper Detection/Integrity Verification

The subsequent steps are followed for the purpose of detecting and locating potential tampering or verifying the integrity of the downloaded image from the cloud or the uploaded image to the cloud, as demonstrated by the activity diagram in Figure 4:

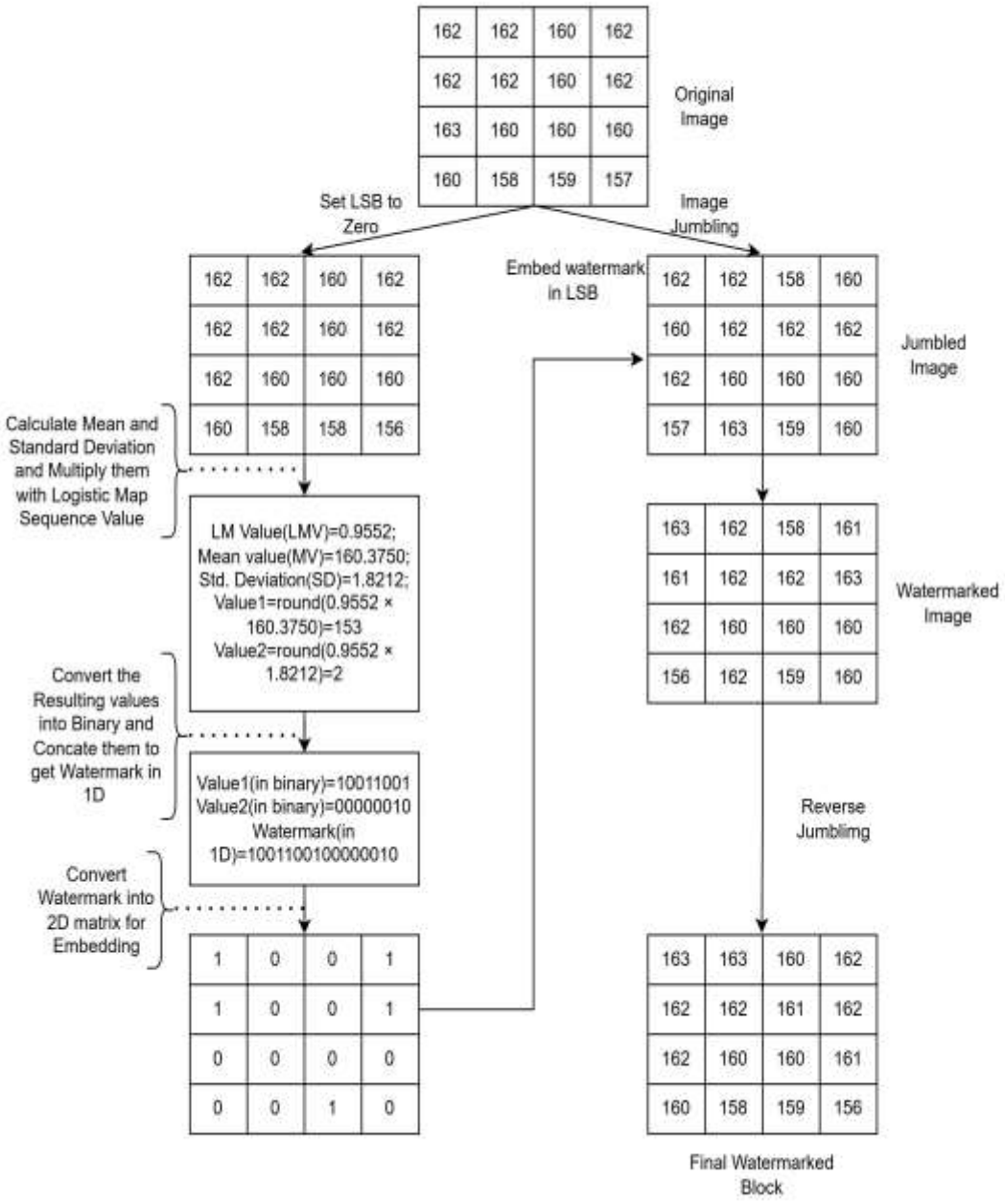

Figure 3: Illustration of Watermark Generation and Embedding with an example

(i) Input the watermarked image.

(ii) Set the watermarked image's least significant bit (LSB) to zero.

(iii) The resulting image is split into 4 × 4 non-overlapping blocks.

(iv) Calculate each block's average and standard deviation. Multiply the obtained result by the same logistic map sequence value used during the embedding process.

(v) Convert the rounded values of resulting values into binary and concatenate them to generate the 16-bit watermark.

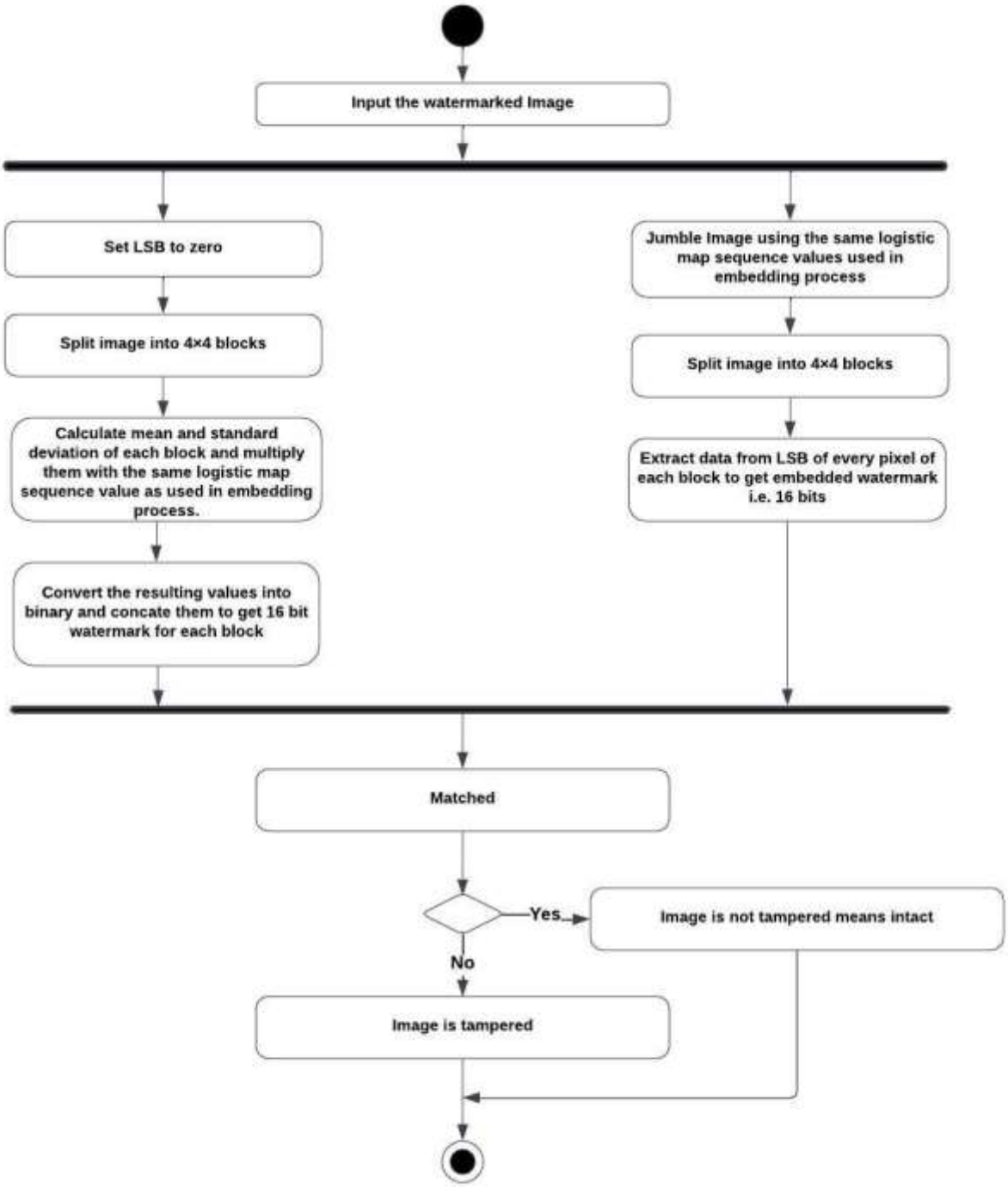

Figure 4: Activity Diagram for Tamper Detection/Integrity Verification

(vi) Jumble the original watermarked image using the same logistic map sequence employed during the embedding process.

(vii) Then, Extract the LSB of every pixel in each block to obtain the embedded 16-bit watermark.

(viii) Compare the calculated watermarked bits for each block (from Step 4) with the extracted watermark bits of the corresponding block (from the previous step) to detect tampering.

(ix) If both sets of bits are identical, it indicates no tampering in the corresponding block, marking it as untampered; otherwise, mark it as a tampered block.

(x) Repeat the previous step for each block. If at least one block is tampered, the integrity of the received image is considered violated; otherwise, it is maintained.

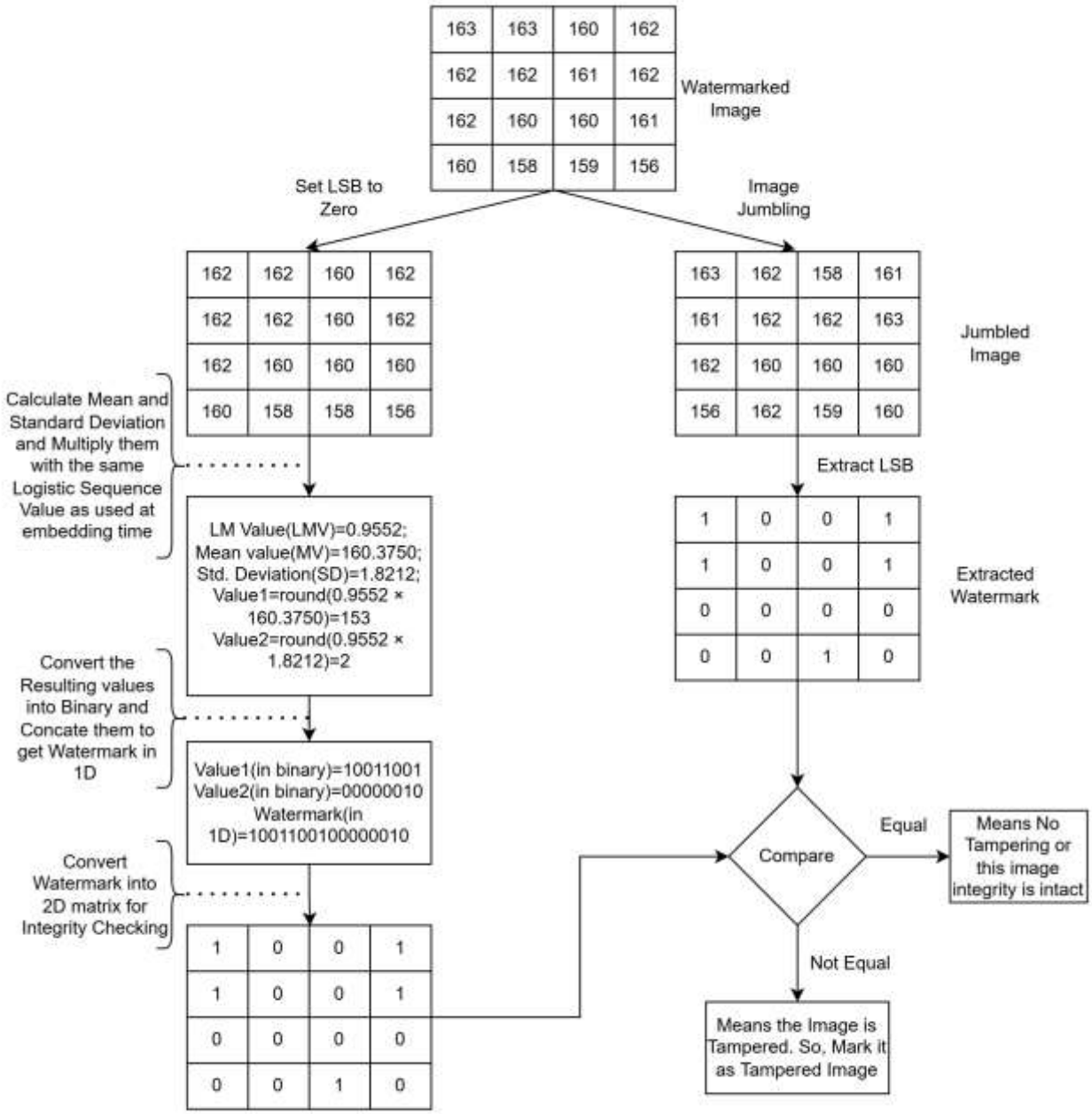

Figure 5: Illustration of Tamper Detection/Integrity Verification with an example

Essentially, the computed watermark undergoes a comparison with the extracted watermark to assess the tampering or integrity in the received image. The entire procedure is illustrated with a straightforward example presented in Figure 5. According to the example, our image remains intact, signifying the absence of tampering, as evidenced by the alignment between the calculated and extracted watermarks. Conversely, a lack of alignment between the two indicates potential tampering with the image.

3. experimental results and discussion

The proposed scheme is evaluated in relation to existing schemes, and a variety of experimental results are presented in this section. All experiments are carried out on an 11th Gen Intel(R) Core (TM) i5-1135G7 (2.40GHz) with 16GB RAM, utilizing MATLAB 2020a on Windows 11. A set of five grayscale sample images that are Lena, cameraman, bungee, onion, pears is used for experimentation that are displayed in Figure 6.

Figure 6: Sample Images for Experiment

To evaluate the proposed scheme, comprehensive quality assessment using PSNR and SSIM is conducted. Additionally, by putting it through a variety of attacks, such as crop, copy-paste, and copy-move attacks, its tamper detection and localization capabilities are evaluated. The computation complexity of watermark embedding and integrity checking was also assessed, considering their execution time. Furthermore, accuracy is also measured using metrics such as FPR, FNR, TDR, among others, as outlined in the following subsections.

3.1 Quality Assessment

The PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index Metric) are essentially the two metrics used to assess the quality and fidelity of a watermarked image relative to the original image. PSNR provides a numerical value in decibels (dB) indicating how much of the original image content is preserved after watermark embedding. It computes the peak signal power to noise power introduced during the watermarking process. A higher PSNR value signifies better image quality after watermark embedding.

On the other hand, three components are used by SSIM to determine how similar the original image and the watermarked image are: luminance, contrast, and structure. These components closely align with human visual perception. The range of SSIM values is -1 to 1, with 1 representing exactly similar.

The proposed scheme is contrasted with existing schemes [5], [4] in terms of average PSNR and SSIM values in Table 1.

Presented in the table, with a PSNR value that is closer to [4], the proposed scheme performs better than [5]. Additionally, the proposed scheme is outperforming than [5] and [4] in case of SSIM value as shown in table 1. Comparing the proposed scheme to [5] and [4], it is clear from the obtained PSNR and SSIM values that it achieves a commendable level of image quality while ensuring higher security.

Table 1: Comparing the Proposed Scheme's average PSNR and SSIM value to Existing Schemes

Metric	[5]	[4]	Proposed Scheme
PSNR	51.1405	51.14922	51.14178
SSIM	0.99642	0.99636	0.9967

3.2 Tamper Detection and Localization Analysis

To assess the integrity verification or tamper detection capabilities of the proposed scheme, variety of watermarked images underwent different attacks such as crop attack, copy-paste attack, and copy-move attack. Subsequently, the proposed scheme was employed to detect and localize these attacks. In the context of this evaluation, a crop attack involves the removal or cropping of a portion of the image, a copy-paste attack involves copying and pasting a portion of an image into a different or identical image and a copy-move attack consists of process of copying and pasting a particular area of an image into same image.

Figure 7a illustrates the detection and localization of a cropping attack on the Lena image. Similarly, Figure 7b showcases the detection of a copy-paste attack, where a region from the Cameraman picture is copied and pasted onto the Bungee image. Additionally, Figure 7c demonstrates the detection of a copy-move attack on the Cameraman image. The tampered watermarked image is always divided into 4×4 size blocks after it is received. Subsequently, the watermark is extracted from the least significant bit (LSB) of each block, and its calculated counterpart is determined using the logistic map, standard deviation, and mean, following the resetting of the LSB to zero, block by block. The comparison of extracted watermark is done with the calculated watermark for each block. Once they match, it indicates no tampering with that block; otherwise, the block is marked as tampered and highlighted. As depicted in Figure 7, the proposed scheme successfully detects and localizes all common attacks on the watermarked image.

(a) Crop Attack

(b) Copy-Paste Attack

(c) Copy-Move Attack

Figure 7: Performance of Proposed Scheme against Different Attacks

Table 2: Comparison of Average Watermark Generation & Embedding Time and Tamper Detection Time of the Proposed Scheme with Already Existing Schemes

Execution Time (in Seconds)	[5]	[4]	Proposed Scheme
Watermark Generation and Embedding Time	1.20466	0.72018	0.18872
Tamper Detection and Localization Time	1.23824	0.25472	0.17424

3.3 Complexity Analysis

The complexity of the proposed scheme can be analysed in the form of execution time. Essentially, the execution time comprises two components: the time required for watermark generation and embedding, and the time dedicated to tamper detection or integrity verification. The average execution time of the proposed scheme is checked against with that of existing schemes, as shown in Table 2, to assess its efficiency.

As analysed in Table 2, the proposed scheme exhibits efficient performance, characterized by significantly lower execution times compared to existing schemes. The reduced execution time can be attributed to the utilization of relatively inexpensive operations such as mean, standard deviation, logistic map, etc., in the watermark embedding and integrity verification processes.

3.4 Tampering Rates Analysis

Further metrics like False Positive Rate (FPR), False Negative Rate (FNR), and Tamper Detection Rate (TDR) are computed to evaluate the performance of the proposed scheme. The formulas for FPR, FNR, and TDR are provided below:

$$FPR = \frac{FP}{(FP + TN)} \tag{2}$$

$$FNR = \frac{FN}{(FN + TP)} \tag{3}$$

$$TDR = \frac{TP}{(TP + FN)} \tag{4}$$

where, the number of tampered pixels that are accurately identified as tampered is called True Positive (TP), TN is the number of non-tampered pixels that have been accurately identified as such, FP represents the count of pixels that are identified as tampered with but are not tampered with, and FN signifies the pixels that are tampered but mistakenly identified as non-tampered. FPR is the rate of falsely detected tampered pixels, FNR is the rate of falsely detected non-tampered pixels, and TDR is the proportion of pixels that are tampered relative to the total number of pixels detected as tampered. TDR is an essential metric for evaluating the proposed scheme's correctness. Table 3 illustrates the average FPR, FNR, and TDR for the proposed scheme in the context of crop, copy-move, and copy-paste attacks across five sample images with varying percentages of tampering. Examining the table reveals that the FNR consistently remains at zero, indicating that the proposed scheme consistently detects tampered pixels. Moreover, the TDR reaches 100%, signifying that all tampered pixels are successfully identified. The False Positive Rate is initially low, but, it gradually increases with higher tampering rates. However, this increase is deemed acceptable when contrasted with the state-of-the-art schemes.

Table 3: Average FPR, FNR and TDR of the Proposed Scheme for Different Tampering Attacks

Tampering Percentage	Crop			Copy Move			Copy Paste		
	FPR	FNR	TDR	FPR	FNR	TDR	FPR	FNR	TDR

10	0.14614	0	100	0.13958	0	100	0.13962	0	100
20	0.1598	0	100	0.15942	0	100	0.15764	0	100
30	0.17786	0	100	0.18234	0	100	0.17958	0	100
40	0.20716	0	100	0.21088	0	100	0.21096	0	100
50	0.25484	0	100	0.24756	0	100	0.25648	0	100

4. CONCLUSION

A fragile digital image watermarking scheme, based on a logistic map, is proposed for integrity verification and authentication of images during transmission between cloud storage and users. The scheme employs a dual level of security by jumbling the image before watermarking. Cost-effective operations such as mean, standard deviation, logistic map, and concatenation are utilized for watermark embedding and tamper detection, improving the proposed scheme to achieve faster execution time.

Moreover, the scheme demonstrates the ability to identify different kinds of attacks on watermarked images, such as crop, copy-paste, and copy-move. Additionally, it maintains good imperceptibility, resulting in high-quality watermarked images, as demonstrated by the values of SSIM and PSNR. Furthermore, the scheme demonstrates significant TDR, FNR and FPR compared to other state-of-the-art schemes.

In the future, the scheme could be extended to incorporate additional features such as enhanced imperceptibility and automatic recovery of tampered regions.

REFERENCES

[1] N. Zermi, A. Khaldi, R. Kafi, F. Kahlessenane, and S. Euschi, "A dwt-svd based robust digital watermarking for medical image security," Forensic science international, vol. 320, p. 110691, 2021.
[2] G. Azizoglu and A. N. Toprak, "A novel reversible fragile watermarking method in dwt domain for tamper localization and digital image authentication," Biomedical Signal Processing and Control, vol. 84, p. 105015, 2023.
[3] S. Gull, N. A. Loan, S. A. Parah, J. A. Sheikh, and G. M. Bhat, "An efficient watermarking technique for tamper detection and localization of medical images," Journal of ambient intelligence and humanized computing, vol. 11, pp. 1799–1808, 2020.
[4] A. K. Sahu, "A logistic map based blind and fragile watermarking for tamper detection and localization in images," Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 8, pp. 3869–3881, 2022.
[5] S. Bhalerao, I. A. Ansari, and A. Kumar, "A secure image watermarking for tamper detection and localization," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 1, pp. 1057–1068, 2021.
[6] S. Bhalerao, I. A. Ansari, and A. Kumar, "A reversible medical image watermarking for roi tamper detection and recovery," Circuits, Systems, and Signal Processing, pp. 1–25, 2023.
[7] A. K. Sahu, M. Hassaballah, R. S. Rao, and G. Suresh, "Logistic-map based fragile image watermarking scheme for tamper detection and localization," Multimedia Tools and Applications, vol. 82, no. 16, pp. 24069–24100, 2023.
[8] M. Fatema, V. Maheshkar, S. Maheshkar, and G. Agarwal, "Tamper detection using fragile image watermarking based on chaotic system," in International Conference on Wireless, Intelligent, and Distributed Environment for Communication: WIDECOM 2018, pp. 1–11, Springer, 2018.
[9] S. Trivedy and A. K. Pal, "A logistic map-based fragile watermarking scheme of digital images with tamper detection," Iranian Journal of Science and Technology, Transactions of Electrical Engineering, vol. 41, pp. 103–113, 2017.
[10] M. Hussan, S. A. Parah, A. Jan, and G. Qureshi, "Hash-based image watermarking technique for tamper detection and localization," Health and Technology, vol. 12, no. 2, pp. 385–400, 2022.
[11] N. J. Megaiz and A. Djebbari, "Securing medical images: A crypto-watermarking and biometrics based scheme," in 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA), pp. 1–5, IEEE, 2022.

[12] S. Sharma, S. Shivani, and N. Saxena, "An efficient fragile watermarking scheme for tamper localization in satellite images," Computers and Electrical Engineering, 2023.
[13] L. Huang, D. Kuang, C.-l. Li, Y.-j. Zhuang, S.-h. Duan, and X.-y. Zhou, "A self-embedding secure fragile watermarking scheme with high quality recovery," Journal of Visual Communication and Image Representation, vol. 83, p. 103437, 2022.
[14] Y. Fang, J. Liu, J. Li, J. Cheng, J. Hu, D. Yi, X. Xiao, and U. A. Bhatti, "Robust zerowatermarking algorithm for medical images based on sift and bandelet-dct," Multimedia Tools and Applications, vol. 81, no. 12, pp. 16863–16879, 2022.
[15] R. Bose and A. Banerjee, "Implementing symmetric cryptography using chaos functions," in Proceedings of the 7th international conference on advanced computing and communications, pp. 318–321, 1999.