Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Improvement of a hamming code and logistic-map based pixel-level active forgery detection scheme

Oussama Benrhouma *

Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah, Saudi Arabia

ARTICLE INFO

Article history:
Received 3 April 2022
Received in revised form
17 June 2022
Accepted 17 June 2022

Keywords:
Cryptanalysis
Improvement
Watermarking tamper
Detection
Chaotic functions
Forgery localization

ABSTRACT

In this paper, we present an improvement of a previously proposed watermarking scheme. We proved in a previous work that the scheme suffers from security flaws and we exploited those flaws to attack the scheme resulting in the revelation of the secret keys used in the embedding process. With possession of the keys watermarked images could be manipulated without being detected by the extraction scheme. In this paper, we propose an improvement of the scheme to cover the security flaws in the scheme. This work falls into the context of improving the design of security systems taking into consideration the different cryptanalysis techniques.

1. Introduction

Information sharing across open networks has grown in tandem with the huge leap in internet technology and online services. As a result, safeguarding and securing sensitive data from unauthorized access is a major problem. Because photos are the most widely utilized medium of information in practically every industry, from social networking to medical, military, biometrics, and banking services, they are the most widely used mode of information. And since digital images could be presented as evidence in courtrooms, concerns about the security of digital images have grown.

Due to redundancy and highly correlated pixels in images, traditional encryption algorithms such as AES, DES, and IDEA are not suitable for image encryption, resulting in inefficiency in confidentiality. As a result, chaos-based security systems became a suitable choice in the design of multimedia security schemes. Ghadirli et al. (2019), Yu et al. (2018), Kaur et al. (2020), and Singh and Sinha (2009), given their properties like high sensitivity to input parameters, random-alike appearance, and ergodicity.

On the other hand, digital signatures are used to control the integrity, the problem is these schemes do not have the ability to locate tampered regions, and since the images are considered huge in size in comparison with texts and since a yes/no answer on integrity question wouldn't be, then these schemes are also not suitable to control the integrity of images, where comes the need to design new schemes taken into consideration the properties of the data in question.

Digital watermarking schemes represent a solution (Benrhouma et al., 2015; 2016; Bravo-Solorio et al., 2018), it consists of embedding some information into the data called "watermark", then the watermark is extracted to serve different goals: In Robust watermarking schemes, the watermark should resist any intentional or unintentional attempt of removal to ensure copy-rights protection (Lu et al., 2006; Simitopoulos et al., 2003; Tang and Hang, 2003; Shahadi et al., 2020). On the other hand, Fragile and semi-fragile watermarking schemes are designed to control the integrity of images, the watermark should be affected by any intentional or unintentional modification of the cover image and that will make locating tampering regions possible (Celik et al., 2002; Chang et al., 2003; Fridrich et al., 2000; Lin and Chang, 2001; Benrhouma et al., 2015; 2016; Di Martino and Sessa, 2012; Haghighi et al., 2019; Bravo-Solorio et al., 2018; Molina-Garcia et al., 2020).

The use of chaotic maps in the design of these schemes will raise its security given that scheme will inherit the chaotic properties and that should make it very difficult to break, but despite that, many cryptanalysis techniques are used to attack these schemes and that result to the cryptanalysis of many

* Corresponding Author.
Email Address: oussama.benrhoumaa@gmail.com
https://doi.org/10.21833/ijaas.2022.09.016
Corresponding author's ORCID profile:
https://orcid.org/0000-0002-4540-3650

chaotic-based security schemes (Benrhouma, 2022; Teng et al., 2013; Botta et al., 2015; Li et al., 2014; Benrhouma et al., 2017; Xie et al., 2017), This proves that relying on the properties of chaotic maps to secure the proposed schemes is not enough, one should take into considerations the cryptanalysis techniques.

In this context, we analyzed the security of a recently proposed watermarking scheme for image forgery detection (Prasad and Pal, 2020), and a successful attack is demonstrated (Benrhouma, 2022), as a result, we were able to calculate the equivalent of the secret keys and manipulate the watermarked images without being detected by the extraction scheme.

In this paper, we continue building on our findings (Benrhouma, 2022) by proposing an improvement of the attacked scheme (Prasad and Pal, 2020). The rest of the paper is organized as follows: Section 2 presents a brief description of the scheme under study, section 3 presents an overview of the executed cryptanalysis, the improvement of the scheme is presented in section 4, and section 5 shows the experimental results, and the paper is concluded in section 6.

2. The scheme under study

The scheme in Prasad and Pal (2020) proposes a fragile watermarking scheme for tamper detection in digital images. The scheme is based on (7,4) hamming code and logistic map: For each pixel, the 4 most significant bits (MSBs) are selected and (7,4) hamming code is used to generate 3-bits authentication code that is then further processed using the logistic map and embedded into the LSBs of the pixel in question. In this section, we present a brief description of the scheme under study.

2.1. Authentication watermark generation and embedding

Given a cover image I with size (M×N) the steps leading to the generation and the embedding of the watermark are as follows:

- Step 1: The logistic map is used to generate a pseudo-random sequence α where $\alpha = \{\alpha i;\ i = 1{:}\ (M \times N)\}$.

The Logistic map is defined by Eq. 1. The values generated by the equations are in [0,1], $\alpha 0$ represents the initial condition provided by the user, and β is the control parameter of the function, where $\beta \in [0, 4]$.

$$\alpha_{i+1} = \beta\alpha_i\ (1 - \alpha_i) \quad (1)$$

The initial condition α0 and the control parameter β are considered as secret keys of the scheme.

- Step 2: At this point, we have a pseudo-random sequence $\alpha = \alpha_i\ (i = 1{:}\ M\ N)$, with the same size as the image, each value from the sequence α will be associated with a pixel, where i represent the index of the pixel in processing.

The pseudo-random sequence is then converted to be in the range from 0 to 7 using Eqs. 2, 3, and 4.

$$Ai = \alpha i \times 255 \quad (2)$$
$$Bi = round(Ai) \quad (3)$$
$$Ki = mod(Bi, 8) \quad (4)$$

- Step 3: The i[th] pixel in the cover image I is selected, converted to binary then its 4 MSBs are selected to compute its hamming code $c = (c7, c6, c5, c4, c3, c2, c1)$.

The watermark is considered the 3 LSBs of the calculated hamming code: $W = (c3, c2, c1)$.

- Step 4: The computed watermark is converted into an integer to obtain T.
- Step 5: Starting from the secret value Ki a list R is created:

$$R = \{Ki, (Ki + 1)\ mod\ 8, (Ki + 2)\ mod\ 8, \ldots, (Ki + 7)\ mod\ 8\} \quad (5)$$

- Step 6: The value of the watermark T is Searched within the list R and its position in R is saved as "j."
- Step 7: Calculate $z = mod\ (Pi, 8)$. Where Pi is the pixel in processing.
- Step 8: Calculate list P R, where $PR = \{Pi - z + t;\ t = 0, 1, 2\ldots, 7\}$.
- Step 9: The watermarked pixel is represented by the j[th] element in the list PR.
- Step 10: The rest of the cover image is processed by applying steps 3 to 9 to obtain the watermarked image WI.

A flowchart of the embedding schemes is shown in Fig. 1.

2.2. Extraction and tamper detection

Given a received watermarked image WI. The steps leading to the extraction of the watermark in order to locate any possible tampering in the image are described as follows:

- Step 1: Generate the same pseudo-random sequence α using the logistic map defined in Eq. 1 with the parameters α_0 and β as secret keys.

α={αi; i=1: (M×N)}

where, (M×N) is the size of the image.

- Step 2: The pseudo-random sequence α is then converted to be in the range from 0 to 7 using Eqs. 2, 3, and 4.

The list K with the same size as the image and each element represents the secret value that will be used to generate the list R for each pixel.

- Step 3: The i[th] pixel PW i in the received image W I is selected, then converted to binary then its 4 MSBs are selected to compute its hamming code:

$$c = (c7, c6, c5, c4, c3, c2, c1)$$

The 3-bits authentication code watermark is the 3 LSBs of the calculated hamming code c:

$W = (c3, c2, c1)$

- Step 4: The list R is generated starting from the elements of the list K: For the i[th] pixel the element Ki is used to calculate the list R:

$$R = \{Ki, (Ki + 1)\ mod\ 8, (Ki + 2)\ mod\ 8, \ldots, (Ki + 7)\ mod\ 8\}$$

- Step 5: Compute $z = mod\ (PW\ i, 8) + 1$ which represents the index of the extracted watermark E_{AC} in the list R.
- Step 6: The comparison between the extracted watermark E_{AC} and the calculated one W will reveal if the pixel in question has been tampered with: each pixel where $E_{AC} \neq W$ is considered falsified, therefore its position in the received image is set to zero which represents the black color.

A flowchart of the extraction and tamper detection schemes is shown in Fig. 2.

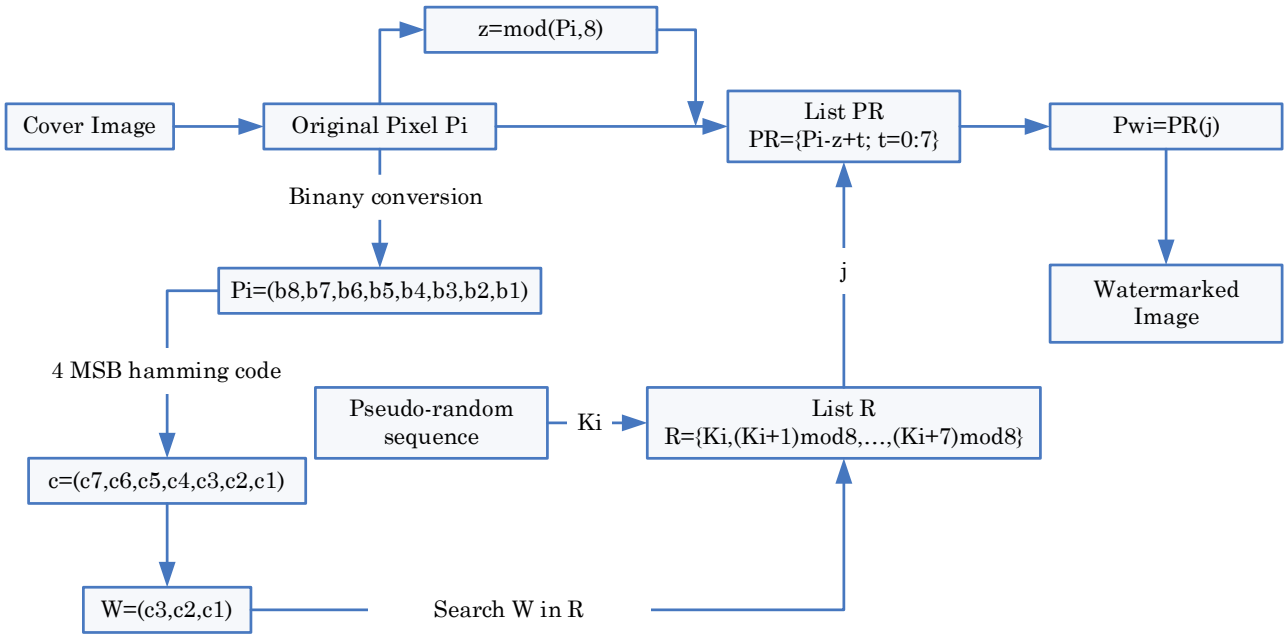

Fig. 1: Flowchart of the embedding scheme (Prasad and Pal, 2020)

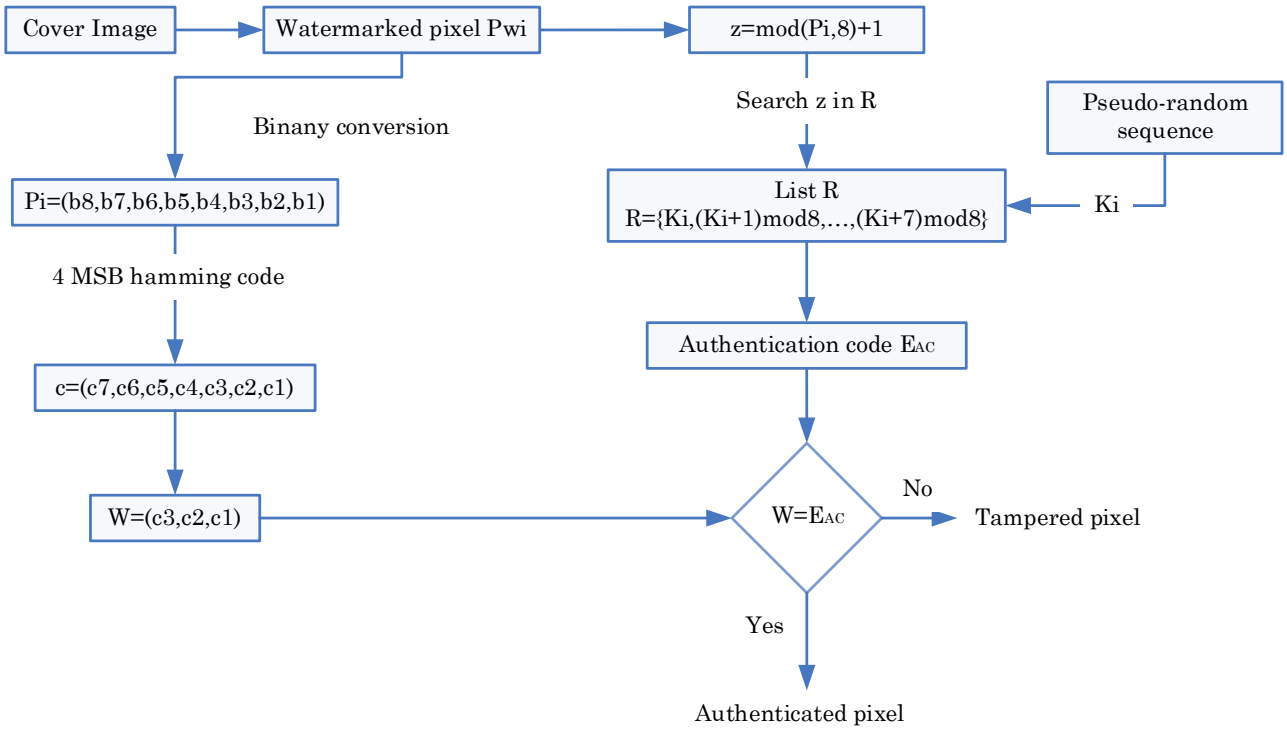

Fig. 2: Flowchart of the extraction scheme (Prasad and Pal, 2020)

3. Cryptanalysis of the scheme

In previous work, we demonstrated in Benrhouma (2022) that the scheme is insecure, two types of attacks were performed (offline and online attack) and the attackers were able to extract and replace the watermark without possession of the keys. As a result of the attack, watermarked images were falsified without being detected by the extraction scheme. In this section, we provide a brief description of the performed cryptanalysis.

The scheme in Prasad and Pal (2020) exploited the chaotic behavior of the logistic map (Eq. 1) to generate pseudo-random sequence α with the same size as the image to be watermarked. The sequence α is then used to construct the list K and R.

The keys of the proposed scheme here are the initial condition of the logistic map α0 and the control parameter β. Given the pseudo-random behavior of the logistic map, and its high sensibility of initial conditions and control parameters, it is nearly impossible to calculate or guess the pseudo-random sequence α, so our goal in the cryptanalysis was to calculate something equivalent to the keys, which in this case, the lists K and R.

We demonstrated with two different types of attacks that the previously mentioned list could be calculated and a successful attack is performed in Benrhouma (2022). As a result, we were able to calculate the lists K and R and use them to embed a new watermark into previously intercepted and falsified watermarked images.

The theoretical calculation was confirmed with experimental results and the extraction scheme failed to detect any manipulations on falsified images.

In this paper, we continue building on our previously mentioned findings (Benrhouma, 2022) by proposing an improvement in the existing scheme (Prasad and Pal, 2020) taking into consideration cryptanalysis techniques. The main problem that we identified in the scheme is the possibility of calculating the sequences K and R. Therefore, we propose a solution to cover this weakness and secure the scheme against similar attacks by exploiting the pseudo-random behavior of the logistic map to encrypt the positions of the watermark.

4. Improvement of the scheme

To cover the weakness of the scheme we propose to add an encryption step on the position of the watermark using pseudo-random generated bits from the logistic map. The encryption will prevent any unauthorized parties from revealing the secret sequence R and therefore protect the scheme from both online and offline attacks.

4.1. Authentication watermark generation and embedding

Given a cover image I with size (M×N) the steps leading to the generation and the embedding of the watermark are as follows:

- Step 1: The logistic map 1 is used to generate a pseudo-random sequence α where $\alpha = \{\alpha i; i = 1: (M \times N)\}$.
- Step 2: The pseudo-random sequence is then converted to be in the range from 0 to 7 using Eqs. 2, 3, and 4.
- Step 3: The i^{th} pixel in the cover image I is selected, converted to binary then its 4 MSBs are selected to compute its hamming code

$$c = (c7, c6, c5, c4, c3, c2, c1)$$

The watermark is considered the 3 LSBs of the calculated hamming code: W=(c3, c2,c1).

- Step 4: The computed watermark is converted into an integer to obtain T.
- Step 5: Starting from the secret value K_i a list R is created:

$$R = \{Ki, (Ki + 1)\ mod\ 8, (Ki + 2)\ mod\ 8, \ldots, (Ki + 7)\ mod\ 8\} \quad (6)$$

- Step 6: The value of the watermark T is searched within the list R and its position in R is saved as "j."
- Step 7: The logistic map is used to generate 3 pseudo-random numbers S, this sequence is then quantified to zeros and ones using threshold L=0.5214. The xor operation is then applied to the sequence S and the saved value j to obtain E.

$$E = j \oplus S$$

- Step 8: Calculate $z = mod\ (Pi, 8)$. Where P_i is the pixel in processing.
- Step 9: Calculate list PR, where $PR = \{Pi - z + t;\ t = 0, 1, 2 \ldots, 7\}$.
- Step 10: The watermarked pixel is represented by the E^{th} element in the list PR.
- Step 11: The rest of the cover image is processed by applying steps 3 to 9 to obtain the watermarked image WI.

A flowchart of the embedding schemes is shown in Fig. 3.

4.2. Extraction and tamper detection

Given a received watermarked image WI. The steps leading to the extraction of the watermark in order to locate any possible tampering in the image are described as follows:

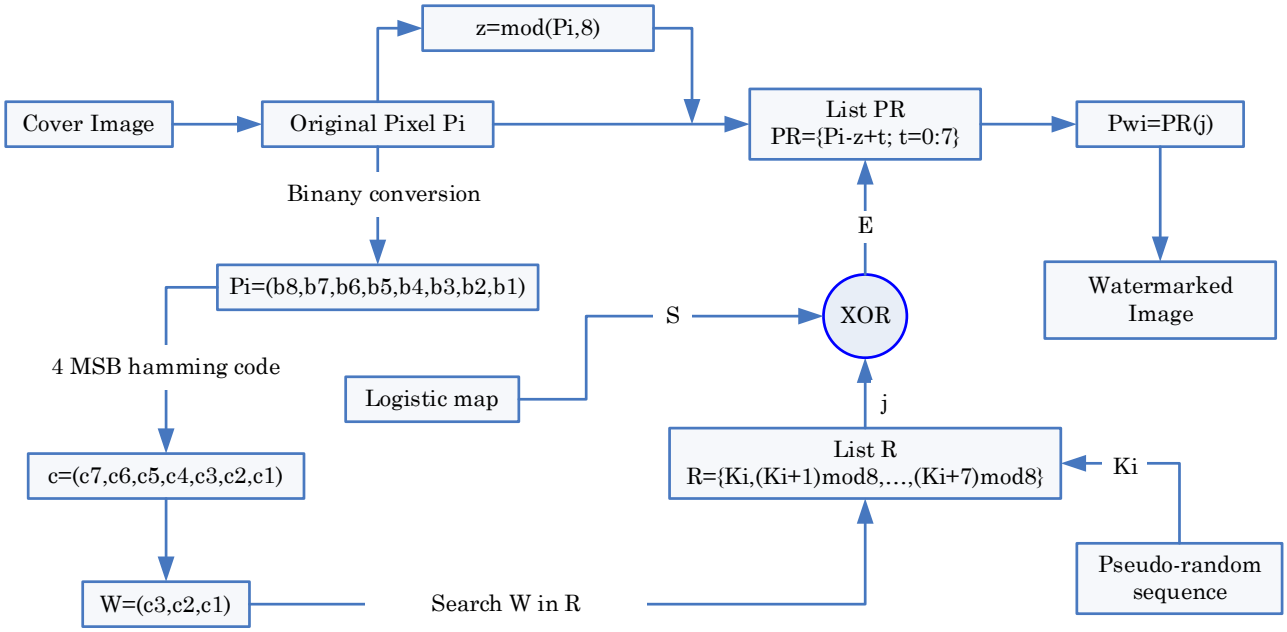

Fig. 3: The improvement of the embedding scheme

- Step 1: Generate the same pseudo-random sequence α using the logistic map defined in Eq. 1 with the parameters α_0 and β as secret keys.

$$\alpha = \{\alpha i;\ i = 1{:}\ (M \times N)\}.$$

where, (M×N) is the size of the image WI.

- Step 2: The pseudo-random sequence α is then converted to be in the range from 0 to 7 using Eqs. 2, 3, and 4.

The list K with the same size as the image and each element represents the secret value that will be used to generate the list R for each pixel.

- Step 3: The i^{th} pixel PW i in the received image WI is selected, then converted to binary then its 4 MSBs are selected to compute its hamming code:

$$c = (c7, c6, c5, c4, c3, c2, c1).$$

The 3-bits authentication code watermark is the 3 LSBs of the calculated hamming code c:

$$W = (c3, c2, c1)$$

- Step 4: The list R is generated starting from the elements of the list K: For the i^{th} pixel the element K_i is used to calculate the list R:

R={Ki, (Ki+1) mod 8, (Ki+2) mod 8, ..., (Ki+7) mod 8,}

- Step 5: Compute $E = mod\ (PWi, 8) + 1$ which represents the encrypted value of the index of the extracted watermark E_{AC} in the list R.
- Step 6: The logistic map is used to regenerate the secret value S using the same parameters, the sequence is then quantified using the threshold L. The position of the watermark in R is revealed by the xor operation between S and E.

$$j = S \oplus E$$
$$R(j) = E_{AC}$$

- Step 7: The comparison between the extracted watermark E_{AC} and the calculated one W will reveal if the pixel in question has been tampered with: Each pixel where $E_{AC} \neq W$ is considered falsified, therefore its position in the received image is set to zero which represents the black color.

A flowchart of the extraction and tamper detection schemes is shown in Fig. 4.

5. Experimental results

In this section we test the perceptual quality of the watermarked images using known metrics, we also test the scheme's performance against various attacks.

5.1. Perceptual quality of watermarked images

Peak Signal-to-Noise Ratio (PSNR): Is used to analyze the visual quality of the watermarked image with comparison to the original image. PSNR is defined by Eq. 7 and the typical values of PSNR are between 30 and 50 dB.

$$PSNR = 10log_{10}\left(\frac{255^2}{MSE}\right) \quad (7)$$

where, MSE is the mean squared error of the original image and the watermarked image given by Eq. 8.

$$MSE = \frac{1}{MN\sum_{i=0}^{M}\sum_{j=0}^{N}(P(i,j)P^{W}(i,j))^2} \quad (8)$$

Structural Similarity (SSIM): Is a method for measuring the similarity of two images. The SSI M index is a metric that measures the image quality based on an initial uncompressed or distortion-free version. SSIM values vary from [0,1], 1 means the two images are totally identical. SSIM is defined by Eq. 9.

$$SSIM = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (9)$$

where, μ_x and μ_y are the average of x and y; σ_x^2 and σ_y^2 are the variances of x and y; σ_{xy} is the covariance of x and y; C1=(k1 L)2 and C2=(k2 L)2 two variables to stabilize the division with a weak denominator; L is the dynamic range of the pixel-values (typically this is $2_{(bitsperpixel-1)}$.); k1=0.01 and k2=0.03 by default.

Table 1 shows the value of PSNR and SSIM between the original and the watermarked images.

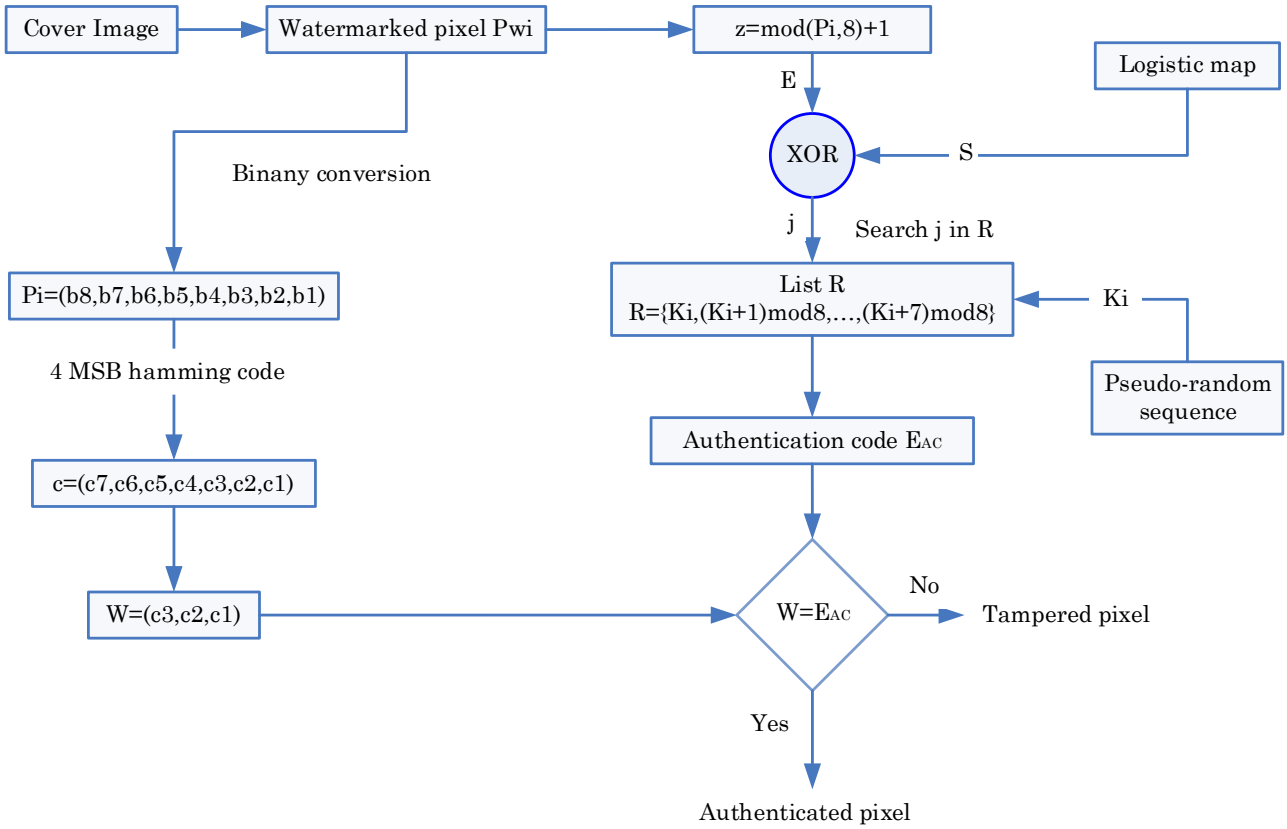

Fig. 4: The improvement of the extraction scheme

Table 1: PSNR and SSIM values for different images

Image	PSNR	SSIM
Lake	37.4211	0.9512
Trucks	37.9850	0.9544
Aerial	37.5462	0.9562
Lena	37.2154	0.9562
Jet	38.0042	0.9688
Tank	38.6522	0.9600

5.2. Tamper detection performance

To evaluate the performance of the improved scheme we calculate the rate of:

- TN (True Negative): Is the number of non-tampered pixels shown true.
- FP (False Positive): Is the number of non-tampered pixels shown as tampered.
- TP (True Positive): Is the number of tampered pixels shown as tampered.
- FN (False Negative): Is the number of tampered pixels shown as not tampered.
- TPR is the True Positive Rate defined by:

$$TPR(\%) = \frac{TP\times100}{TP+FN} \quad (10)$$

- TNR is the True Negative Rate defined by:

$$TNR(\%) = \frac{TN\times100}{TN+FP} \quad (11)$$

- FPR is the False positive Rate defined by:

$$FPR(\%) = \frac{FP\times100}{TN+FP} \quad (12)$$

- FNR is the False Negative Rate defined by:

$$FNR(\%) = \frac{FN\times100}{TP+FN} \quad (13)$$

- ρ is the tampering ratio, defined by:

$$\rho(\%) = \frac{(FN+TP)\times100}{M\times N} \quad (14)$$

where, (M×N) is the size of the image.

5.3. Effectiveness of the improved scheme against various attacks

In this section we analyze the effectiveness of the improved scheme against various attacks:

- Copy-paste attack: In the copy-paste attack the watermarked image is modified by adding parts to the image which are copied from the same

watermarked image. The results of this attack are shown in Fig. 5.

- Collage attack: The collage attack is made by adding some elements in the watermarked image which have been copied from some other watermarked image. The results of this attack are shown in Fig. 6.
- Text-addition attack: The text addition attack simply consists of adding some text to the watermarked image. The results of this attack are shown in Fig. 7.
- Object deletion attack: In this experiment, parts or objects of the image are deleted. The results show that the watermarking scheme is effective to detect such attacks. The experiment is illustrated in Fig. 8.

In all the previously mentioned experiments, the algorithm was able to locate positively the tampered regions. The results are shown in Figs. 5, 6, 7, and 8. Full quantitative statistics of these experiments are drawn in Table 2. It should be noted that in Table 2:

- SSIM is the Structural Similarity between the original and watermarked image.
- PSNR is the Peak Signal-to-Noise Ratio between the original and watermarked image.
- Tampered is the actual number of tampered pixels.
- Not tampered is the actual number of non-tampered pixels.

(a) aerial

(b) Watermarked aerial
SSIM = 0.9562
PSNR = 37.5462

(c) Manipulated aerial
ρ = 1.6647%

(d) Tampering map
TPR = 99.9083%
TNR = 99.5810%

Fig. 5: Effectiveness against copy-paste attack

(a) Watermarked Trucks
PSNR = 0.9544
SSIM = 37.9850

(b) Watermarked Tank
PSNR = 0.9600
SSIM = 38.6522

(c) Manipulated aerial
ρ = 7.1198%

(d) Tampering map
TPR = 100%
TNR = 99.0439%

Fig. 6: Effectiveness against collage attack

(a) Jet

(b) Watermarked jet
SSIM = 0.9688
PSNR = 38.0042

(c) Manipulated aerial
ρ = 0.9933%

(d) Tampering map
TPR = 100 %
TNR = 99.4813%

Fig. 7: Effectiveness against text-addition attack

(a) Lake

(b) Watermarked lake
SSIM = 0.9512
PSNR = 37.4211

(c) Manipulated aerial
ρ = 2.6062%

(d) Tampering map
TPR = 99.8829%
TNR = 98.5143%

Fig. 8: Effectiveness against object deletion attack

Table 2: Tamper detection performance

	Aerial	Trucks	Jet	Lake
TP	1090	4666	651	1706
TN	64175	60288	64516	63518
FP	270	582	369	310
FN	1	0	0	2
FPR %	0.4190	0.9561	0.5687	0.4857
FNR %	0.0907	0	0	0.1171
TPR %	99.9083	100	100	99.8829
TNR %	99.5810	99.0439	99.4813	99.5143
ρ %	1.6647	7.1198	0.9933	2.6062
Tampered	1091	4666	651	1708
Non tampered	64445	60870	64885	63828
SSIM	0.9562	0.9544	0.9688	0.9512
PSNR	37.5462	37.9850	38.0042	37.4211

5.4. Comparison with related work

In this section, the improved version is compared with some of the existing techniques. in terms of its ability to detect and locate possible forgeries, the ability of recovering the tampered regions and in terms of the mean values of PSNR and SSIM of watermarked images.
The proposed scheme showed better PSNR values of watermarked images and similar SSIM values. The results are shown in Table 3.

Table 3: Comparison with related work

	Tamper local-ization	Recovery	Mean PSNR	Mean SSIM
Benrhouma et al. (2015)	Yes	Yes	42	0.9654
Shojanazeri et al. (2017)	Yes	No	41	0.9677
Proposed	Yes	No	38	0.9578

6. Conclusion

In this paper, an improvement of a previously attacked watermarking scheme is proposed, we proposed to encrypt the positions from the list where the watermark is selected. The encryption is performed by generating pseudo-random numbers using the logistic map, these numbers are then quantified to zeros and ones using a secret threshold and the xor operation is performed to confuse the position of the selected watermark. This step will prevent any possible attacker from identifying the position and revealing the lists which are considered equivalent to the keys of the cryptosystem. An encryption scheme should take into consideration the possibility of attacks, this work comes to complete our previous successful attack performed on this very watermarking scheme (Benrhouma, 2022) and to prove that using pseudo-random functions in the design of these schemes doesn't guarantee its security.

Acknowledgment

The authors would like to thank the deanship of research at the Islamic University of Madinah for supporting this research.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Benrhouma O (2022). Cryptanalysis of a hamming code and logistic-map based pixel-level active forgery detection scheme. International Journal of Advanced Computer Science and Applications, 13(2): 663-668. https://doi.org/10.14569/IJACSA.2022.0130278

Benrhouma O, Hermassi H, and Belghith S (2015). Tamper detection and self-recovery scheme by DWT watermarking. Nonlinear Dynamics, 79(3): 1817-1833. https://doi.org/10.1007/s11071-014-1777-3

Benrhouma O, Hermassi H, and Belghith S (2017). Security analysis and improvement of an active watermarking system for image tampering detection using a self-recovery scheme. Multimedia Tools and Applications, 76(20): 21133-21156. https://doi.org/10.1007/s11042-016-4054-2

Benrhouma O, Hermassi H, El-Latif A, Ahmed A, and Belghith S (2016). Chaotic watermark for blind forgery detection in images. Multimedia Tools and Applications, 75(14): 8695-8718. https://doi.org/10.1007/s11042-015-2786-z

Botta M, Cavagnino D, and Pomponiu V (2015). A successful attack and revision of a chaotic system based fragile watermarking scheme for image tamper detection. AEU-International Journal of Electronics and Communications, 69(1): 242-245. https://doi.org/10.1016/j.aeue.2014.09.004

Bravo-Solorio S, Calderon F, Li CT, and Nandi AK (2018). Fast fragile watermark embedding and iterative mechanism with high self-restoration performance. Digital Signal Processing, 73: 83-92. https://doi.org/10.1016/j.dsp.2017.11.005

Celik MU, Sharma G, Saber E, and Tekalp AM (2002). Hierarchical watermarking for secure image authentication with localization. IEEE Transactions on Image Processing, 11(6): 585-595. https://doi.org/10.1109/TIP.2002.1014990 **PMid:18244657**

Chang EC, Kankanhalli MS, Guan X, Huang Z, and Wu Y (2003). Robust image authentication using content based compression. Multimedia Systems, 9(2): 121-130. https://doi.org/10.1007/s00530-003-0083-6

Di Martino F and Sessa S (2012). Fragile watermarking tamper detection with images compressed by fuzzy transform. Information Sciences, 195: 62-90. https://doi.org/10.1016/j.ins.2012.01.014

Fridrich J, Goljan M, and Baldoza AC (2000). New fragile authentication watermark for images. In the International Conference on Image Processing, IEEE, Vancouver, Canada: 446-449. https://doi.org/10.1109/ICIP.2000.900991

Ghadirli HM, Nodehi A, and Enayatifar R (2019). An overview of encryption algorithms in color images. Signal Processing, 164: 163-185. https://doi.org/10.1016/j.sigpro.2019.06.010

Haghighi BB, Taherinia AH, and Mohajerzadeh AH (2019). TRLG: Fragile blind quad watermarking for image tamper detection and recovery by providing compact digests with optimized quality using LWT and GA. Information Sciences, 486: 204-230. https://doi.org/10.1016/j.ins.2019.02.055

Kaur G, Agarwal R, and Patidar V (2020). Chaos based multiple order optical transform for 2D image encryption. Engineering Science and Technology, an International Journal, 23(5): 998-1014. https://doi.org/10.1016/j.jestch.2020.02.007

Li M, Zhang J, and Wen W (2014). Cryptanalysis and improvement of a binary watermark-based copyright protection scheme for remote sensing images. Optik, 125(24): 7231-7234. https://doi.org/10.1016/j.ijleo.2014.07.130

Lin CY and Chang SF (2001). A robust image authentication method distinguishing JPEG compression from malicious manipulation. IEEE Transactions on Circuits and Systems for Video Technology, 11(2): 153-168. https://doi.org/10.1109/76.905982

Lu W, Lu H, and Chung FL (2006). Robust digital image watermarking based on subsampling. Applied Mathematics and Computation, 181(2): 886-893. https://doi.org/10.1016/j.amc.2006.02.012

Molina-Garcia J, Garcia-Salgado BP, Ponomaryov V, Reyes-Reyes R, Sadovnychiy S, and Cruz-Ramos C (2020). An effective fragile watermarking scheme for color image tampering detection and self-recovery. Signal Processing: Image Communication, 81: 115725. https://doi.org/10.1016/j.image.2019.115725

Prasad S and Pal AK (2020). Hamming code and logistic-map based pixel-level active forgery detection scheme using fragile watermarking. Multimedia Tools and Applications, 79(29): 20897-20928. https://doi.org/10.1007/s11042-020-08715-x

Shahadi HI, Thahab AT, and Farhan HR (2020). A novel robust approach for image copyright protection based on concentric rectangles. Journal of King Saud University-Computer and Information Sciences, 34(4), 1263-1274. https://doi.org/10.1016/j.jksuci.2020.06.006

Shojanazeri H, Wan Adnan WA, Syed Ahmad SM, and Rahimipour S (2017). Authentication of images using Zernike moment watermarking. Multimedia Tools and Applications, 76(1): 577-606. https://doi.org/10.1007/s11042-015-3018-2

Simitopoulos D, Koutsonanos DE, and Strintzis MG (2003). Robust image watermarking based on generalized radon transformations. IEEE Transactions on Circuits and Systems for Video Technology, 13(8): 732-745. https://doi.org/10.1109/TCSVT.2003.815947

Singh N and Sinha A (2009). Optical image encryption using Hartley transform and logistic map. Optics Communications, 282(6): 1104-1109. https://doi.org/10.1016/j.optcom.2008.12.001

Tang CW and Hang HM (2003). A feature-based robust digital image watermarking scheme. IEEE Transactions on Signal Processing, 51(4): 950-959. https://doi.org/10.1109/TSP.2003.809367

Teng L, Wang X, and Wang X (2013). Cryptanalysis and improvement of a chaotic system based fragile watermarking scheme. AEU-International Journal of Electronics and Communications, 67(6): 540-547. https://doi.org/10.1016/j.aeue.2012.12.001

Xie EY, Li C, Yu S, and Lü J (2017). On the cryptanalysis of Fridrich's chaotic image encryption scheme. Signal Processing, 132: 150-154. https://doi.org/10.1016/j.sigpro.2016.10.002

Yu C, Li J, Li X, Ren X, and Gupta BB (2018). Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. Multimedia Tools and Applications, 77(4): 4585-4608. https://doi.org/10.1007/s11042-017-4637-6