



ISSN: 0067-2904

Using Visual Cryptography and hash function for Fragile Watermarking to Detect Electronic Document Forgery

Hala k. Hussein^{1*}, Ra'ad A. Muhajjar¹, Bashar S Mahdi²

¹ Department of Computer Science, College of Computer Science and Information Technology, Basrah, Iraq

² Department of Computer Science, College of Computer Science, University of Technology, Baghdad, Iraq

Received: 16/8/2022

Accepted: 19/10/2022

Published: 30/7/2023

Abstract

Recent developments in technology and the area of digital communications have rendered digital images increasingly susceptible to tampering and alteration by persons who are not authorized to do so. This may appear to be acceptable, especially if an image editing process is necessary to delete or add a particular scene that improves the quality the image. But what about images used in authorized governmental transactions? The consequences can be severe; any altered document is considered forged under the law and may cause confusion. Also, any document that cannot be verified as being authentic is regarded as a fake and cannot be used, inflicting harm on people. The suggested work intends to reduce fraud in electronic documents using a novel generator. It employs modern encryption methods to conceal the data represented by the watermark and hide it secretly inside the document so that it is easy to check for signs of counterfeiting; thus, allowing for the discovery of forged papers

Keywords: Fragile watermark, spatial domain, LSB, government Document, PNG

استخدام التشفير المرئي ووظيفة التجزئة للعلامة المائية الهشة لكشف تزوير المستندات الإلكترونية

هاله خالد حسين^{1*}, رعد عبد الحسن مهجر¹, بشار سعدون مهدي²

¹ قسم علوم الحاسوب، كلية علوم الحاسوب وتكنولوجيا المعلومات، جامعة البصرة، بصرة، العراق

² قسم علوم الحاسوب، كلية علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق

الخلاصة

التطورات الأخيرة في التكنولوجيا ومجال الاتصالات جعلت الصور الرقمية أكثر عرضة للتلاعب والتعديل من قبل الأشخاص غير مصرح لهم بذلك. للوهلة الأولى، قد يبدو هذا جيدًا، خاصةً إذا كانت عملية تحرير الصورة ضرورية لحذف أو إضافة مشهد معين يحسن جمال الصورة. ولكن ماذا عن الصور المستخدمة في المعاملات الحكومية المصرح بها؟ فكر في العواقب: أي مستند تم تغييره يعتبر مزورًا بموجب القانون وقد يؤدي إلى إرباك. وأي وثيقة لا يمكن التحقق من أنها ذات أصول قانونية تعتبر مزورة ولا يجوز استخدامها، مما يلحق الضرر بمزايا المواطن. يهدف العمل المقترح إلى تقليل الاحتيال في المستندات الإلكترونية باستخدام مولد جديد.

*Email: itpg.hala.khalid@uobasrah.edu.iq

يستخدم طرق التشفير الحديثة لإخفاء البيانات التي تمثلها العلامة المائية وإخفائها سرّاً داخل المستند حتى يسهل التحقق من علامات التزوير؛ وبالتالي، سيتم التحقق من صحة الأوراق المزورة واكتشافها ضمن هذا العمل.

1. Introduction

Recently, it has become easier than ever to obtain any digital information, especially with the development and variety of programs on the internet, which has facilitated sending and receiving images and other multimedia between the sender and several other receiving parties[1], [2]

Thus, image security is now a major problem. There are several approaches for preserving digital images. Watermarking and encrypted methods are only two examples of the various options available for preventing unauthorized changes to images[3]. A watermark is a collection of bits representing secret information that is added to an image to prevent unauthorized usage. To ensure that the watermark does not affect the image, it should be embedded into the image rather than placed separately. The changes to the image when adding a watermark would be nearly unrecognizable to the naked eye, yet computers can still analyze it[4]. In other words, digital watermarking might be visible or invisible[5], but this work would be invisible.

Moreover, a watermark might be robust, fragile, or semi-fragile[6]. For example, if an image needs to be secured against malicious and unintentional attacks. In this situation, robust watermarking should be used, whereas semi-fragile watermarking is best for combating content-protected alterations like Gaussian noise and JPEG compression. Alterations to fragile watermarking are easily noticeable[7]. Two steps make up the watermarking process: embedding and extracting[8], [9]. The watermark can be extracted from the image for various purposes during extraction. Depending on extraction type, this approach may be categorized into three distinct groups: blind, non-blind, and semi-blind watermarking[10]. Furthermore, it is applied in spatial or frequency domains or hybrid systems[11].

2. Related works

Several watermark algorithms have been developed to check the content of forged images. Raj & Shreelekshmi in [12], has suggested the use of two different types of fragile watermarks in order to improve security. The host image was first divided into eight 8x8 sections. MD5 was then used to produce a 128-bit representation. Two least significant bits (2LSB) were used to embed the original fragile watermark in each block. This second method is quite similar to the first, with the exception that it divides the picture into 16x16 blocks. The SHA-256 technique was used to produce a watermark, which generates and embed in the least significant bits of an original image.

In [13], Gul & Ozturk, suggested that the Spatial Domain, watermarks can be embedded using LSB. They used SHA-1 hash technique to analyze three blocks of each image divided into four parts to generate a watermark. To determine if a watermark was tampered with or genuine, it would be extracted and compared to a hash. This authentication was less secure due to being particularly vulnerable to pixel-based attacks like salt and paper noise.

Molina-Garcia et al., in [14], suggested using a color image to create three recovery watermarks for the host picture by splitting each color into 4*4 blocks, extracting 6 bits from each block, and then embedding the three watermarks for detection on the tampered region. They proposed employing an inpainting technique that helps restore blocks deleted due to manipulation or coincidental problems. They also advise utilizing the 2LSB approach to preserve picture quality and applying hierarchal tamper detection to achieve high detection

accuracy when extracting watermarks, separating each channel into 4*4 blocks and then using the 2LSB to display chrominance and authentication watermarks.

Ayu et al. [15], proposed technique provided dual-layer fragile digital watermarking, where two watermarks are embedded in medical images. They employed Advanced Encryption Standard. (AES) to encrypt the Electronic Patient Record (EPR) for confidentiality and authenticity. Then, the Secure Hash Algorithm (SHA256) was used to validate Digital Imaging and Communication in Medicine (DICOM) tags' integrity after embedding them in 2LSB as the initial fragile watermark. Finally, they separated the image into non-overlapping blocks, assigned each one an id, then calculated the SHA256 for integrity and employed it as a tamper detector

Shen et al. in [16], employed Singular Value Decomposition (SVD) to generate a fragile watermark that offers authentication information. They divided non-overlapping blocks into two sections: upper and lower. Combining the informational components of authentication from the bottom and top portions creates an authentication code by simply inserting these codes using 2 LSB into each block. Researchers could identify whether a block was tampered with or not after collecting the authentication information for each block. The problem with this method is that it can't deal with image compression. As a result, it cannot determine if an image was compressed using Vector Quantization or JPEG (VQ).

In [17], Reyes-Reyes et al., proposed a schema for RGB color images, slicing the image to non-overlapping blocks. Then, generating the recovery and authentication watermarks from each block by applying Pseudocode to each channel. In this stage, there will be three watermarks that could embed and provide more security—in the second stage, the XOR procedure on each recovery watermark will generate a single bit for the block authentication process. They also solved the TCP problem by designing a framework called (SR-HTR) with a painting process.

3. Proposed System General Architecture

This proposal covers three stages: In this initial step, explain the generated watermark. The second step involves embedding the watermark, which must be done via a specific technique. Finally, the receiver would extract the image watermark and compare it to the original. Color image pixels are converted to RGB images. Each color in this image included eight bits of Red, eight bits of Green, and eight bits of Blue since it is in a 24-bit format. Alternately, each color is composed of two parts: Most Significant Bit (MSB) and the Least Significant Bit (LSB), which are four bits.

The project relied entirely upon this spatial domain, which was used to change pixel values without degrading image quality. A lot of information needed to be encoded and using color images increased the embedding capacity since each color has a set of bits that are used to give a pixel value.

Values are embedded into 2LSB, where each pixel's bit should be replaced with the matching watermark bit, as seen in the Table (1).

Table 1: Shows 2LSB technique

Color value in the image	Secret information	MSB				4LSB	3LSB	2LSB	1LSB	Value after replacement
RED	1	1	1	1	0	0	1	0→1	0	230
GREEN	1	1	0	0	0	0	1	0→1	1	135
BLUE	0	0	1	1	0	0	0	1→0	0	104

i. Watermark generator

Due to the dangers of information leakage and manipulation, it is necessary to use a watermark generator by combining visual cryptography and hashing to secure its images from any of those vulnerabilities. Additionally, watermark generators minimize the number of bits concealed in a cover image by discarding duplicate bits. This paper proposes algorithms to achieve those goals. Creating the secret bit of the watermark can be achieved in the following steps:

A. Proposed Visual Cryptography

The proposed approach employs Visual Cryptography (VC) to generate two random images from a secret image. For the highest level of protection and authenticity of the owner's watermark, it's ideal to use two different versions of the file, one for private use and one for public consumption.

A public shared image is usually used for creating a watermark. In contrast, a private image is maintained in the watermark extraction process to guarantee that an attacker cannot determine the location of a concealed image by analyzing the original Logo. The recommended VC strategy consists of a secret encode table (distribution phase), where the two secret images are stacked or overlapped, then merged and rebuild for each pixel to look exactly like it did when it was first created, and a visual cryptography process (reconstruction phase), where the two shared images are stacked or overlapped, then merge and rebuild for each pixel until it looks like it did when it was first created.

In the suggested work, VC and hash are used in the generation stage. The user-specified logo is used by VC to produce bits that are divided into two sharing images. When matching the images, one is inserted into the image while the other is added to the program. The logo is used to detect fraud or legitimacy. According to Figure 1, the logo in this image is distributed equally between private and public shares, each one has 128 bit in size.

B. Hash function

The hash value are calculated from the operation of the hash. As soon as it is obtained, follow these steps sequentially:

Step 1: The hash function takes three different inputs in the proposed system the; input will be (Department, Year, Word). This input could change according to the sender and receiver.

Step 2: from the previous step, the hash will generate 64 digits, and the whole hash function output will be broken up into eight pieces; each code will include eight characters.

Step 3: the eight characters are transformed in every block into binary values so each code block will contain 32 bits

Step 4: this binary number is provided to the embedding operation so that it can be inserted at the location that is chosen as shown in Figure 1

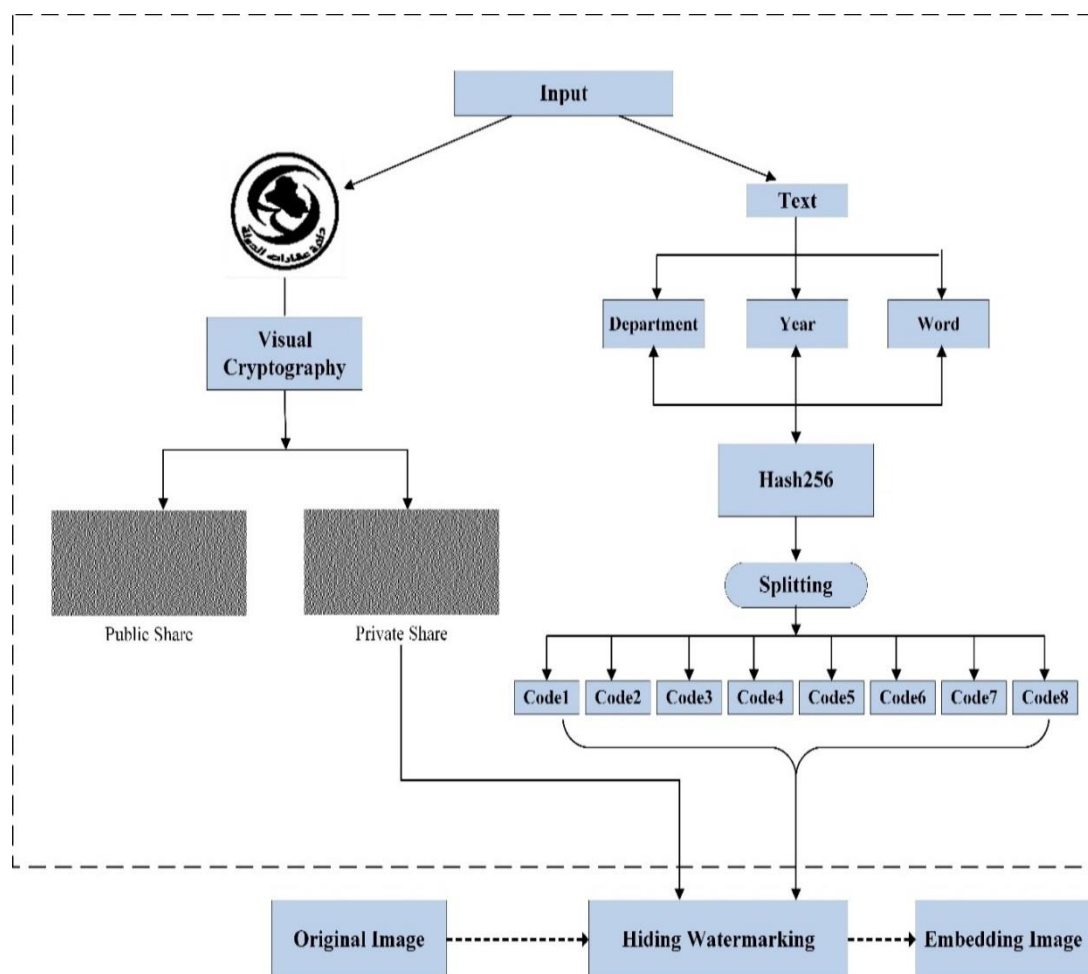


Figure 1: Watermark Generator structure

ii. Watermark Embedding Stage:

This stage involves dividing the image into eight blocks of uniform size and embedding a particular amount of confidential data from the watermarking generation step into each block. How much information should be in a block is discussed in the next section:

1. Fetch the image in PNG format.
2. Divided the image into eight blocks
3. Select one block to insert secret information.
4. Calculate the secret information obtained from both (hash output and public share from visual cryptography).
5. Select a pixel from the block.
6. Split each pixel into three original colors (RED, Green, Blue).
7. Select one color of the three colors.
8. Calculate the sequence of color and convert its value to binary
9. Insert one bit of secret information into each color of the pixel.
10. Use 2LSB in step8 of inserting the bits
11. Repeat all from step 4 until all colors have one bit taken from the secret information
12. The Secret information will start decreasing until its value becomes zero, and then the process will repeat on another block until the eighth block is completed. Figure 2 shows all these steps.

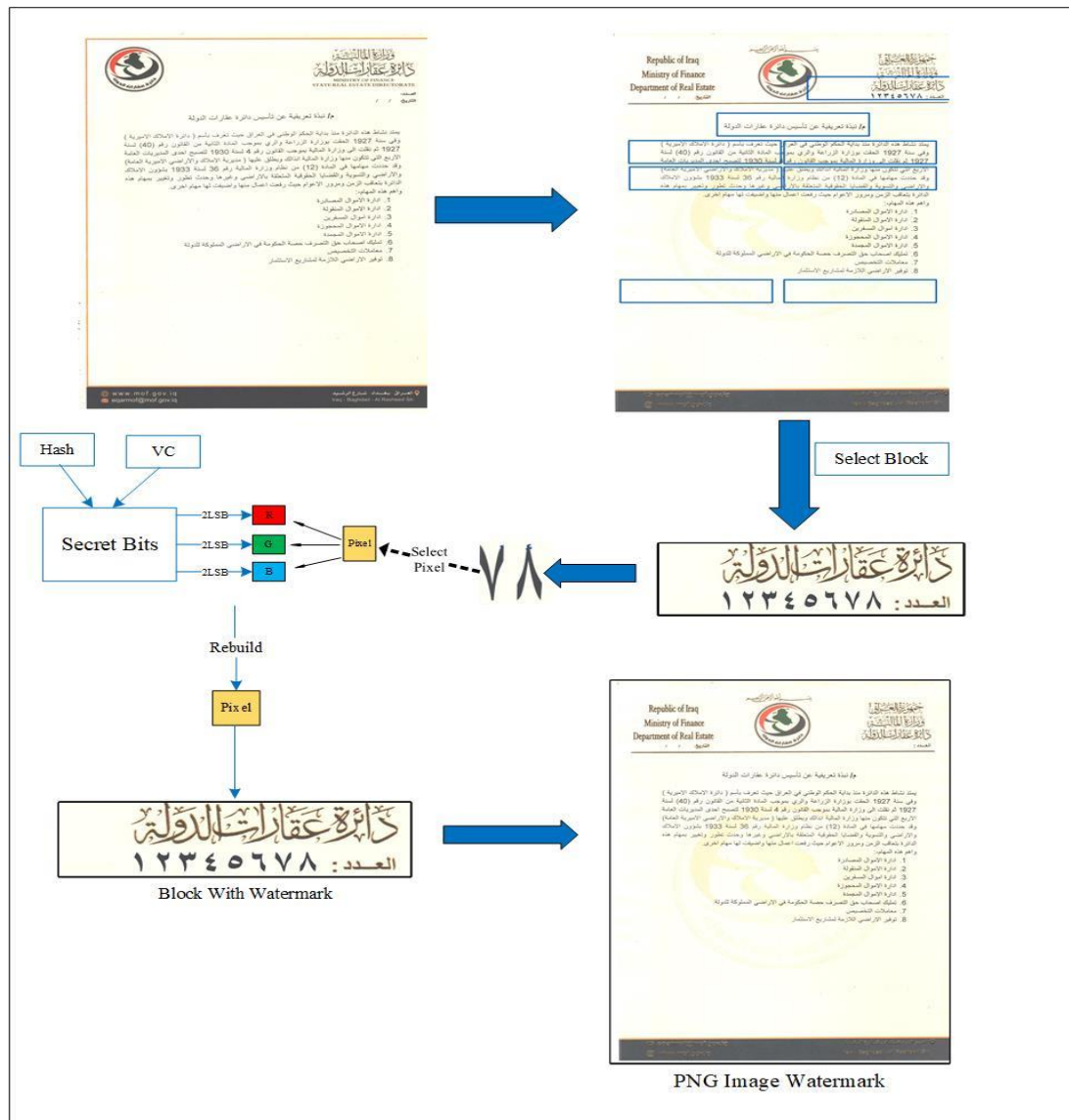


Figure 2: Watermark embedding stage

iii. Watermark Extraction Stage:

1. Test the image that contains the watermark.
2. Select a block to start the process of extraction.
3. Do the same procedure in the embedding process from step4 to step7.
4. Extract one bit of secret information from each color separately (RED, GREEN, BLUE).
5. Select 2LSB that we want to extract.
6. Combine all the bits from each pixel.
7. Store the extracted bits from each pixel into a specific variable.
8. Repeat all steps from 3 to 7 until all secret information is extracted.
9. Divided the secret information into two parts. One for the bits hash function output, and the other for public share VC.
10. Reconstruct the VC and bits of hash function and compare it with the original watermark. If it matches, that means the image is not tampered with; else it would be considered a forgery. All these steps are shown in Figure3.

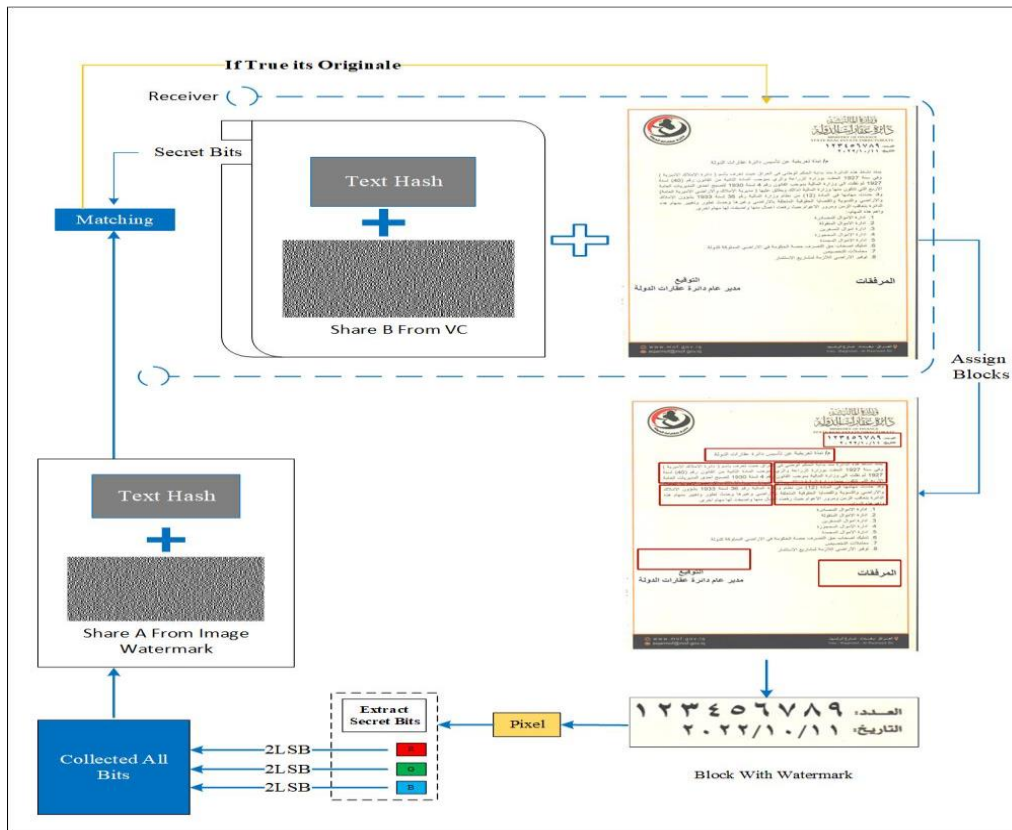


Figure 3: Watermark Extraction Stage

4. Implemented Measures

The security of the suggested technique can be evaluated using several metrics, including the Peak Signal Noise Ratio, Mean Square Error, Bit Error Rate, Histogram Analysis, and others.

1- Mean Square Error (MSE)

It shows the exact square difference between the images' un-watermarked and watermarked versions. The following equation [1] description of MSE It shows the exact square difference between the un-watermarked and watermarked versions of the image. In general, MSE has no precise value, although the lower the value, the better, and zero are optimal[18].

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{N \times M} (C_i - S_i)^2 \quad [1]$$

While $M \times N$ is the size of the image, C_i and S_i are, respectively, watermark and host images. a main problem with MSE is its dependency on the intensity of pixel

2- Imperceptibility

Regarding watermarks, imperceptibility is one of the most important aspects to consider. As a result, the amount of noise introduced to the picture must be quantified by the watermark bits to assess the image's overall quality. PSNR is used for this purpose. This unit of measurement is referred to as an image quality metric. Increasing the ratio will improve image quality, which is the most important factor. The equation [2] of PSNR can be defined as [19].

$$PSNR = 20 \times \log_{10}(MAX) - 10 \times \log_{10}(MSE) \quad [2]$$

Max is the highest pixel value that may be achieved, while MSE is the mean square error.

3- Bit Error Rate (BER)

BER, defined as the rate at which transmission system mistakes occur, which allows for testing a system's actual performance while it is in use. The bit error rate (BER) offers the best means of achieving system integrity. The following equation [3] can be used to determine the bit error rate [20]:

$$\text{BER} = \text{number of errors} / \text{total number of bits sent} \quad [3]$$

5. Results of Experiment

The supplied form has been executed in the scope of this proposed work on three photos, which are government papers of the image type (PNG) format for the State Real Estate Department linked with the Ministry of Finance, with (900, 1238) where the numbers in parentheses are the size of the worksheet, width, and height consecutively. Table (2) and Figure 4 will include models with a histogram for documents before and after the embedding process. To authenticate papers and identify any manipulation, the fragile watermark should be employed, by using a technique that has been clarified earlier

The major purpose of the suggested framework is not only to construct a watermark and embed it inside an image, but this mark must also be fragile to identify any manipulation that happens on the image, which means that it will be destroyed once an alteration occurs. This manipulation is considered forging if it isn't within the acceptable range.

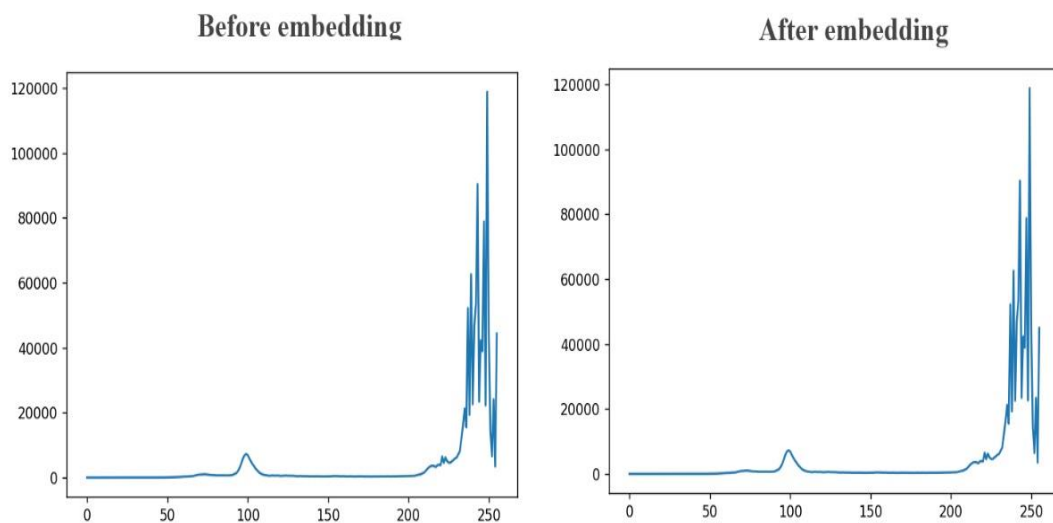



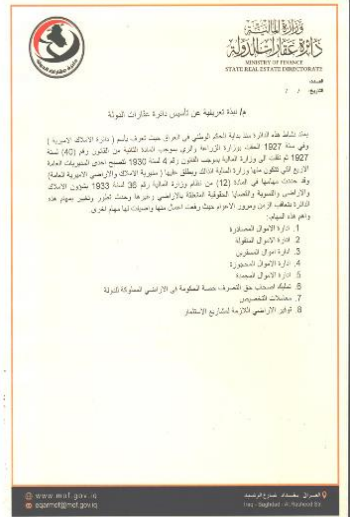
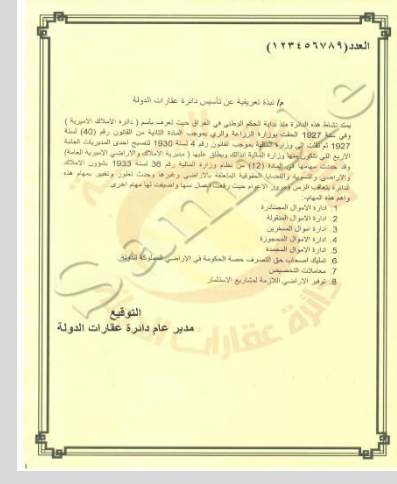
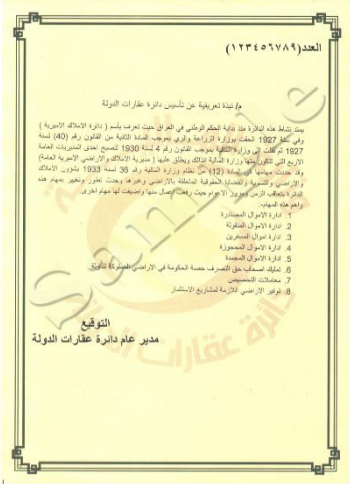


Figure 4: Show A Histogram for The Original PNG Image And a Watermarked Image.

Table 2: shows the PNG Image Before and After Insert Watermark.





Images	PNG image	image with a watermark	PSNR	MSE
Test1			65.45DB	0.01
Test2			65.47DB	0.018
Test3			65.44DB	0.018

The VC serves as a threshold for determining whether an image is fraudulent. This VC is created by combining two photos to ensure that the authentication is genuine and not fabricated. Additionally, the hash function's benefits have been used by giving one-way outputs, providing that this procedure had greater integrity.

Some picture assaults was performed to validate the suggested system's operation and influence the image's quality. Since the image consists of PNG-formatted government documents, the fundamental need for such an important work was that it be in its standard form, i.e., not reverse-rotated, not cropped, and of sufficient quality to enable a continuation of reading the document's contents. The attacks implemented were Rotate, salt and pepper, scale.

Measurements are shown in the Table (3) below. Following the attacks, the image was rotated and checked to see whether the original bits could be recovered once they returned to their original shape. It was able to overcome this assault. On the contrary, in scale attacks, there was no way to extract the watermark because it was too fragile, and the pixel values changed when zoomed in or out, affecting the extraction process. As for the Salt and Pepper assault, the watermark was successful, and most of the bits were recovered.









Table 3: shows The Result After an Attack on an Image with a Fragile Watermark.

TEST1	Attacks	PSNR	MSE	LOGO
	Rotate (90, -90)	65.50 DB	0.01	
	Scale	38.94 DB	8.28	
	Salt and Pepper	44.90 DB	2.10	
	Blur filter	36.08 DB	16.02	

6. Detecting Forgery Block

Several attacks were carried out in this space to assess the watermark's vulnerability and to determine what blocks image forgery. It is possible to modify an image in Photoshop by doing operations such as copy-pasting, adding text, copying-moving a portion of the image to a new spot, or adding a new component, as Table 4 displays. Our results indicate that the system successfully detected this block alteration. BER might be used as a threshold to determine whether blocks succeeded or failed.

Table 4: Detecting forgery blocks

Name Of Attack	Extracted Logo	Hash Blocks Table																											
 <p>Original document with watermark.</p>	 <p>Original extracted logo.</p>	<table border="1"> <thead> <tr> <th>Index blocks</th> <th>ratio</th> <th>result</th> </tr> </thead> <tbody> <tr><td>1</td><td>0</td><td>Pass</td></tr> <tr><td>2</td><td>0</td><td>Pass</td></tr> <tr><td>3</td><td>0</td><td>Pass</td></tr> <tr><td>4</td><td>0</td><td>Pass</td></tr> <tr><td>5</td><td>0</td><td>Pass</td></tr> <tr><td>6</td><td>0</td><td>Pass</td></tr> <tr><td>7</td><td>62</td><td>Fail</td></tr> <tr><td>8</td><td>0</td><td>Pass</td></tr> </tbody> </table>	Index blocks	ratio	result	1	0	Pass	2	0	Pass	3	0	Pass	4	0	Pass	5	0	Pass	6	0	Pass	7	62	Fail	8	0	Pass
Index blocks	ratio	result																											
1	0	Pass																											
2	0	Pass																											
3	0	Pass																											
4	0	Pass																											
5	0	Pass																											
6	0	Pass																											
7	62	Fail																											
8	0	Pass																											
<p>Insert Element from A Different Image</p>  <p>Document with horizontal line inserted.</p>	 <p>Logo with horizontal line inserted.</p>	<table border="1"> <thead> <tr> <th>Index blocks</th> <th>ratio</th> <th>result</th> </tr> </thead> <tbody> <tr><td>1</td><td>0</td><td>Pass</td></tr> <tr><td>2</td><td>0</td><td>Pass</td></tr> <tr><td>3</td><td>56</td><td>Fail</td></tr> <tr><td>4</td><td>0</td><td>Pass</td></tr> <tr><td>5</td><td>0</td><td>Pass</td></tr> <tr><td>6</td><td>0</td><td>Pass</td></tr> <tr><td>7</td><td>0</td><td>Pass</td></tr> <tr><td>8</td><td>0</td><td>Pass</td></tr> </tbody> </table>	Index blocks	ratio	result	1	0	Pass	2	0	Pass	3	56	Fail	4	0	Pass	5	0	Pass	6	0	Pass	7	0	Pass	8	0	Pass
Index blocks	ratio	result																											
1	0	Pass																											
2	0	Pass																											
3	56	Fail																											
4	0	Pass																											
5	0	Pass																											
6	0	Pass																											
7	0	Pass																											
8	0	Pass																											
<p>A Similar Image Component Has Been Applied</p>  <p>Document with horizontal lines applied.</p>	 <p>Logo with horizontal lines applied.</p>	<table border="1"> <thead> <tr> <th>Index blocks</th> <th>ratio</th> <th>result</th> </tr> </thead> <tbody> <tr><td>1</td><td>0</td><td>Pass</td></tr> <tr><td>2</td><td>0</td><td>Pass</td></tr> <tr><td>3</td><td>0</td><td>pass</td></tr> <tr><td>4</td><td>46</td><td>Fail</td></tr> <tr><td>5</td><td>0</td><td>Pass</td></tr> <tr><td>6</td><td>0</td><td>Pass</td></tr> <tr><td>7</td><td>0</td><td>Pass</td></tr> <tr><td>8</td><td>0</td><td>pass</td></tr> </tbody> </table>	Index blocks	ratio	result	1	0	Pass	2	0	Pass	3	0	pass	4	46	Fail	5	0	Pass	6	0	Pass	7	0	Pass	8	0	pass
Index blocks	ratio	result																											
1	0	Pass																											
2	0	Pass																											
3	0	pass																											
4	46	Fail																											
5	0	Pass																											
6	0	Pass																											
7	0	Pass																											
8	0	pass																											
<p>Write on image</p>  <p>Document with text overlay.</p>	 <p>Logo with text overlay.</p>	<table border="1"> <thead> <tr> <th>Index blocks</th> <th>ratio</th> <th>result</th> </tr> </thead> <tbody> <tr><td>1</td><td>0</td><td>Pass</td></tr> <tr><td>2</td><td>0</td><td>Pass</td></tr> <tr><td>3</td><td>0</td><td>pass</td></tr> <tr><td>4</td><td>46</td><td>Fail</td></tr> <tr><td>5</td><td>0</td><td>Pass</td></tr> <tr><td>6</td><td>0</td><td>Pass</td></tr> <tr><td>7</td><td>0</td><td>Pass</td></tr> <tr><td>8</td><td>0</td><td>pass</td></tr> </tbody> </table>	Index blocks	ratio	result	1	0	Pass	2	0	Pass	3	0	pass	4	46	Fail	5	0	Pass	6	0	Pass	7	0	Pass	8	0	pass
Index blocks	ratio	result																											
1	0	Pass																											
2	0	Pass																											
3	0	pass																											
4	46	Fail																											
5	0	Pass																											
6	0	Pass																											
7	0	Pass																											
8	0	pass																											

7. A Comparison of Several Alternative Systems

The proposed method was compared to previous techniques to determine if image quality is maintained after embedding. There is a difference in how the watermark bits are generated, but the embedding process is in 2LB. Results showed a higher percentage of PSNR than other methods, as demonstrated in Table 5

Table 5: shows a comparison with other methods.

Method	PSNR (DB)
N. R. N. Raj and R. Shreelekshmi	51.13
E. Gul and S. Ozturk	57.15
Molina-Garcia et al.,	44.13
Ayu et al.	44.91
Shen et al.	46.36
Reyes-Reyes et al.	43.89
Proposed Method	65.47

8. Conclusion

This research study presented a way for generating a watermark and embedding it in the images to check its originality or whether it has been altered. The Watermark proved its fragility in many manipulations and succeeded in overcoming other attacks that may occur on paper, such as noise that usually occurs to the image due to several causes. In addition to the process of embedding in the second bit, it contributed to the additional image fragility necessary to achieve the study's objective. Furthermore, the use of a lossless compression method, like PNGs, helped preserve the picture quality after embedding, with a PSNR percentage of 65.47 and MSE 0.018, which is acceptable.

With the use of hashing and visual cryptography methods, this study contributed to developing a novel generator for the fragile Watermark, which offered security in that the attacker did not know what the inputted Watermark was. Therefore, the method utilized in this study may be used for more sensitive photographs and documents, such as government files, to identify any modifications on these documents. In addition to the Watermark's natural fragility in this approach, the proposed method is effective in maintaining image quality.

References

- [1] A. Alsimry, K. Hussein Ali, and E. Wahab Abood, "A new approach for finding duplicated words in scanned Arabic documents based on OCR and SURF," *J. Basrah Res.*, no. 47, 2021, Accessed: Jul. 07, 2022. [Online]. Available: <https://www.iasj.net/iasj/journal/260/issues>.
- [2] I. Q. Abduljaleel and A. H. Khaleel, "Significant medical image compression techniques: A review," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 5, pp. 1612–1621, 2021, doi: 10.12928/TELKOMNIKA.v19i5.18767.
- [3] I. Q. Abduljaleel, "Using IWT and LSB Method to Hide Encrypted image in Color image," *Journal of Basrah Researches*, pp. 1–16, 2016.
- [4] H. M. Abdul-Nabi, E. S. Al-Shawi, and H. L. Hussain, "Hiding Three Images at one image by Using Wavelet Coefficients at Color Image," *Basrah J. Sci.*, 2010.
- [5] I. I. Hamid and E. Muzaffer Jamel, "Image Watermarking using Integer Wavelet Transform and Discrete Cosine Transform," *J. Sci.*, vol. 57, no. 2B, pp. 1308–1315, 2016.
- [6] D. Singh and S. K. Singh, "DCT based efficient fragile watermarking scheme for image authentication and restoration," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 953–977, 2017, doi: 10.1007/s11042-015-3010-x.
- [7] S. H. Mnkash and M. E. Abdulmunem, "A Review of Software Watermarking," *Iraqi J. Sci.*, vol. 61, no. 10, pp. 2740–2750, Oct. 2020, doi: 10.24996/IJS.2020.61.10.30.
- [8] Z. J. Ahmed and L. E. George, "Robust Watermarking for Video Using Mean Modulation Technique," *J. Sci.*, vol. 58, no. 4C, pp. 2412–2426, 2017, doi: 10.24996/ij.s.2017.58.4C.17.
- [9] D. R. Alshibani and Z. Sadeq, "Image Content Verification based on DWT and Chaotic Map Watermarking," *Iraqi J. Sci.*, vol. 59, no. 1C, pp. 607–616, Mar. 2018, doi: 10.24996/ij.s. 2018.59.1C.18.
- [10] M. Lebcir, S. Awang, and A. Benziane, "Robust blind watermarking technique against geometric

- attacks for fingerprint image using DTCWT-DCT,” *Iraqi Journal of Science*, vol. 61, no. 10. pp. 2715–2739, 2020, doi: 10.24996/ij.s.2020.61.10.29.
- [11] M. M. Laftah, “3D Model Watermarking based on Wavelet Transform,” *Iraqi J. Sci.*, vol. 62, no. 12, pp. 4999–5007, Dec. 2021, doi: 10.24996/IJS.2021.62.12.36.
- [12] N. R. N. Raj and R. Shreelekshmi, “Blockwise Fragile Watermarking Schemes for Tamper Localization in Digital Images,” *2018 International CET Conference on Control, Communication, and Computing, IC4 2018*. pp. 441–446, 2018, doi: 10.1109/CETIC4.2018.8530950.
- [13] E. Gul and S. Ozturk, “A novel hash function based fragile watermarking method for image integrity,” *Multimedia Tools and Applications*, vol. 78, no. 13. pp. 17701–17718, 2019, doi: 10.1007/s11042-018-7084-0.
- [14] J. Molina-Garcia, B. P. Garcia-Salgado, V. Ponomaryov, R. Reyes-Reyes, S. Sadovnychiy, and C. Cruz-Ramos, “An effective fragile watermarking scheme for color image tampering detection and self-recovery,” *Signal Process. Image Commun.*, vol. 81, p. 115725, Feb. 2020, doi: 10.1016/J.IMAGE.2019.115725.
- [15] M. A. Ayu, T. Mantoro, and I. M. A. Priyatna, “Advanced watermarking technique to improve medical images’ security,” *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 17, no. 5, pp. 2684–2696, 2019, doi: 10.12928/TELKOMNIKA.v17i5.13292.
- [16] J. J. Shen, C. F. Lee, F. W. Hsu, and S. Agrawal, “A self-embedding fragile image authentication based on singular value decomposition,” *Multimed. Tools Appl.*, 2020, doi: 10.1007/s11042-020-09254-1.
- [17] R. Reyes-Reyes, C. Cruz-Ramos, V. Ponomaryov, B. P. Garcia-Salgado, and J. Molina-Garcia, “Color image self-recovery and tampering detection scheme based on fragile watermarking with high recovery capability,” *Applied Sciences (Switzerland)*, vol. 11, no. 7. 2021, doi: 10.3390/app11073187.
- [18] O. Evsutin, A. Melman, and R. Meshcheryakov, “Digital steganography and watermarking for digital images: A review of current research directions,” *IEEE Access*, vol. 8, pp. 166589–166611, 2020, doi: 10.1109/ACCESS.2020.3022779.
- [19] L. Rakhmawati, Wirawan, and Suwadi, “Image Fragile Watermarking with Two Authentication Components for Tamper Detection and Recovery,” *2018 International Conference on Intelligent Autonomous Systems, ICoIAS 2018*. pp. 35–38, 2018, doi: 10.1109/ICoIAS.2018.8494080.
- [20] M. R. T. Dilshad Mahjabeen, “BIT ERROR RATE ANALYSIS IN DIFFERENT TERRAINS FOR LTE,” *Int. J. Res. -GRANTHAALAYAH*, 2019, Accessed: Sep. 29, 2022. [Online]. Available: https://www.granthaalayahpublication.org/journals/granthaalayah/article/view/IJRG19_A01_2050/647.