

PERFORMANCE OF IMAGE WATERMARKING ALGORITHM USING RSA WITH FIBONACCI TRANSFORM AGAINST ATTACKS

ZAHARI MAHAD¹, NUR RAIDAH SALIM^{1,*}, KAMILAH ABDULLAH²,
SUHAILA ABDUL HALIM², NURUL AINA SYAFIQAH ZULKAFLI²

¹Institute for Mathematical Research, University Putra Malaysia,
43400 UPM Serdang, Selangor DE, Malaysia

²College of Computing, Informatics and Media, Al-Khwarizmi Building,
Universiti Teknologi MARA, 40450 Shah Alam, Selangor DE, Malaysia

*Corresponding Author: nurraidah@upm.edu.my

Abstract

The protection of digital information has been the centre of attention as it can be transferred via the Internet, which may allow irresponsible parties to exploit the flaws in security. One such technique to protect digital information is digital image watermarking. This article introduces a watermarking algorithm using Discrete Wavelet Transform and Modified RSA Cryptosystem to secure the digital image. A secret image, which is to be protected, is embedded into a cover image. Several attacks are applied to the watermarked image to test the robustness of the embedded secret image. The metrics used to evaluate its performance are the peak signal-to-noise ratio and structural similarity index measure. Finally, it is compared with existing methods.

Keywords: Digital image watermarking, DWT, Imperceptibility, Robustness, RSA cryptosystem.

1. Introduction

Digital image watermarking has gained traction in recent years as it turns relevant in many applications, such as authentication, broadcast monitoring, and copy control. The application of digital watermarking is involved with many topics such as signal processing, cryptography, theory of probability, stochastic theory, and design of algorithms [1]. The watermarking process includes embedding and extracting the secret image on a cover image.

A watermarking algorithm can be categorized as non-blind or blind [2]. A non-blind algorithm requires the original cover image without any embedded secret image to carry out the extraction process. In contrast, a blind algorithm does not need it to extract the embedded secret image. In the design of an image watermarking algorithm, the important parameters are imperceptibility, robustness, embedding capacity, security, and efficiency. Identifying the reliability of a good algorithm proves challenging because improving on one or more parameters negatively affects another parameter [3].

To ensure that the embedded watermark or secret image is robust, it needs to be embedded in the transform domain. An image is said to be in a spatial domain and can be transformed into the transform domain using certain methods of transformation. One such transformation is the discrete wavelet transform (DWT). Embedding the secret image in the DWT domain offers great robustness against various attacks [4, 5].

The security of the secret image also needs to be considered. For this purpose, encryptions are applied. Since algorithms are often made public, secret keys for decryption are important so that the secret image is secure, even when the method of embedding is known. Watermarking algorithms often implement scrambling techniques to scramble the pixel values of an image in a chaotic manner and get a seemingly meaningless image as a result. The secret key could be either the number of times the scrambling is applied to the image, or the parameters used in the scrambling formula. It is recommended to use an efficient encryption technique requiring less scrambling or a secure one requiring more than one key [6]. To make it even more secure, a cryptosystem can be applied along with the image scrambling as an extra layer of security.

Numerous studies have invented the cryptosystem into the watermarking system, addressing security concerns. For instance, applying the RSA asymmetric encryption algorithm to guarantee the security of the hidden data has proved to lower the consuming time and improve the watermark's security apart from ensuring the high robustness of the watermarked image [7]. Distinct keys are generated for encryption and decryption under an asymmetric architecture combined with a fractional chaotic system that employs fast computing power multiplication, which improves security and encryption effects [8]. Another study used the Rabin-p cryptosystem to secure the watermark and enhance the computational complexity by encrypting the important side data. Also, descramble the scrambled image with Arnold's cat map [9]. Moreover, the suggested watermarking technique exhibits a high level of robustness against the ten attacks, as evidenced by the significant values of NC towards colour and binary secret image [10].

This article proposes a non-blind, robust and imperceptible image watermarking algorithm applying DWT, Fibonacci transform, and Modified RSA Cryptosystem.

The secret image embedded in the DWT domain is robust and imperceptible, while the Fibonacci transform encrypts the secret image by scrambling it. On top of that, a cryptosystem, Modified RSA Cryptosystem [11] is also applied for image encryption. The intervention of the Modified RSA Cryptosystem has not been proven anywhere in the context of digital image watermarking and is expected to give new insight into this area.

There are four sections following this introduction section. The Theoretical Background section explains the techniques used in the proposed algorithm, as well as how they are used in the algorithm. The Methodology section explains the steps of the proposed algorithm. The Results and Discussions section contains the performance evaluation of the proposed algorithm. Lastly, the Conclusions section summarizes the whole article and includes suggestions for future works.

2. Theoretical Background

This section briefly details the techniques used in the proposed algorithm. The techniques used are DWT, Fibonacci transform, and Modified RSA Cryptosystem. The Fibonacci transformation encrypts the secret image by scrambling it into a meaningless image to protect it. Lastly, the Modified RSA Cryptosystem is applied to further encrypt the image and requires a secret key to correctly decrypt and restore the scrambled image. The subsections are arranged in order of the steps of the proposed algorithm.

2.1. Fibonacci transform

The Fibonacci Transform is a scrambling technique based on the Fibonacci sequence, which is generalized as in Eq. (1) [12].

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \begin{pmatrix} F_i & F_{i+1} \\ F_{i+2} & F_{i+3} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \pmod{en} \quad (1)$$

where $x, y \in \{0, 1, 2, \dots, N-1\}$, (x_i, y_i) is the coordinate of a pixel, (x_i', y_i') is the scrambled position of that pixel, $N \in \{0, 1, 2, \dots, N-1\}$ is the size of the digital image, and F_i is the i -th term of the Fibonacci series. Each pixel of the image will be moved to a different position until all pixels have been moved. An example of the first matrix FT_i is shown below [12].

$$FT_1 = \begin{pmatrix} F_1 & F_{1+1} \\ F_{1+2} & F_{1+3} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$$

The process is done on every pixel, which will be moved to a unique position. The process is reversible by carrying out the transformation again. This scrambling is carried out on the cover and secret images.

2.2. Modified RSA cryptosystem

The Modified RSA Cryptosystem [11] is a modified version of the RSA algorithm based on the number of used prime numbers. Compared to the original RSA algorithm, this version of the cryptosystem has the advantage of added security and no loss of data on images [11]. Like most cryptosystems, it has three processes, which are key generation, encryption, and decryption. Each of the processes is detailed in Algorithms 1, 2, and 3 [11].

<p>Algorithm 1 Modified RSA Cryptosystem Key Generation</p> <p>Input: Three primes p, q, r Output: Public key (e, n) and secret key d</p> <ol style="list-style-type: none"> 1. Select three large prime numbers p, q, r to form $n = pqr$. 2. Compute $t = (p-1)(q-1)(r-1)$ 3. Choose (e) such that $1 < e < t$. 4. Find d such that $ed = 1 \pmod{t}$ 5. Announce (e, n) as the public key. 6. Keep d as the secret key.
--

Before the key is generated, the three large primes $p, q,$ and r and variable e need to be chosen. The public key (e,n) is used for encryption and the secret key d is being used decryption processes.

<p>Algorithm 2 Modified RSA Cryptosystem Encryption Algorithm</p> <p>Input: Plain image $P,$ public key (e, n) Output: Cipher image C</p> <ol style="list-style-type: none"> 1. Read the plain image P into its corresponding matrix 2. Partition W into sub block $i*i$ call it S_p. 3. Reshape each sub block into a vector $(1, i*i)$ and call it u_2 4. Compute $C = u^e \pmod{n}$ by computing element by element such that $Ci = ui^e \pmod{n}$. 5. Reshape each Ci to sub block $i*i$ that is denoted by S_c. 6. Construct the cipher image C by gathering the sub blocks S_c such that every subblock S_c is the corresponding sub block to S_p in the plain image.

Once the image has been encrypted, knowing the secret key is needed to correctly decrypt the cipher image.

<p>Algorithm 3 Modified RSA Cryptosystem Decryption Algorithm</p> <p>Input: Cipher image $C,$ public key $(e, n),$ private key d Output: Plain image P</p> <ol style="list-style-type: none"> 1. Read the cipher image into its corresponding matrix 2. Divide into sub block $i*i$ call it S_{ci}. 3. Reshape each sub block into a vector $(1, i*i)$ and call it u_2 4. Apply the decryption algorithm $P_d = (u_2)^d \pmod{n}$ over u_2 by computing element by element such that $P_{di} = (u_{2i})^d \pmod{n}$ 5. Reshape each P_{di} to sub block $i*i$ that is denoted by S_d. 6. Construct the decrypted image gathering the sub blocks S_d such that every sub block S_d is the corresponding sub block to S_{ci} in the cipher image.
--

Once the decryption is done, the original image is restored. In the proposed algorithm, the Modified RSA Cryptosystem is used to encrypt the scrambled secret image to obtain the encrypted-scrambled secret image. The double encryption will ensure that it is protected from attackers on the watermarked image.

2.3. Discrete wavelet transform domain

The DWT process transforms an image from the spatial domain to the transform domain, where the embedded watermark can be more robust and imperceptible.

The DWT will decompose an image into four sub-bands, LL, LH, HL, and HH, as illustrated in Fig. 1.



Fig. 1. DWT image transform.

Each sub-bands have half the dimensions of the original cover image. For the proposed algorithm, the LL sub-band is chosen for secret image embedding due to its advantages in robustness [13]. The LL sub-band is divided into 8x8 sub-blocks where the secret image is to be embedded.

3. Proposed Method

This section details on the steps of the proposed watermarking algorithm. There are two processes in the algorithm, which are embedding and extraction. The algorithm is then subject to performance evaluation to better understand its strengths and weaknesses. Four cover images and one secret image in .jpeg format and size 512x512 pixels were used. Figure 2 shows the cover and secret images used for watermarking and its evaluation.

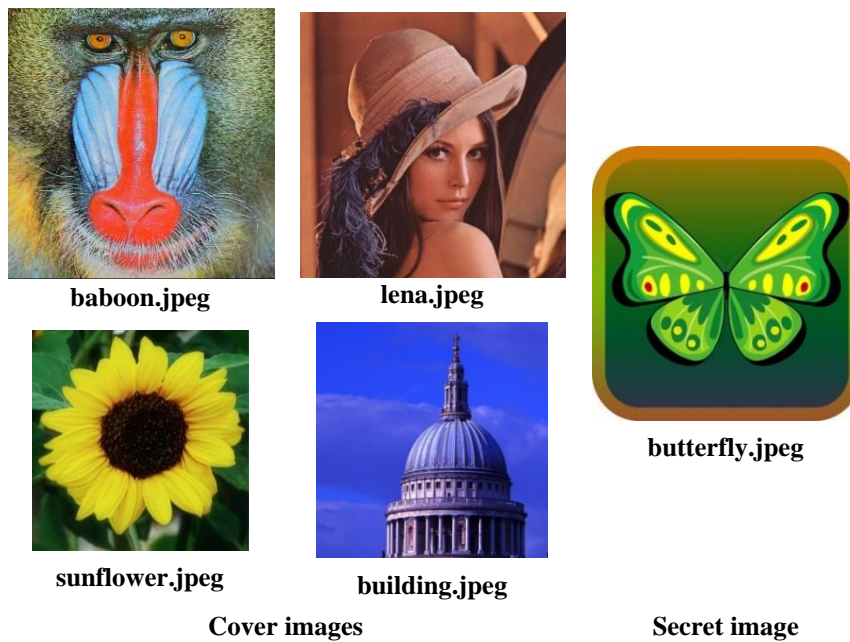


Fig. 2. Test cover images and a secret image.

3.1. Embedding process

The embedding process is where the secret image is inserted into the cover image, so the output of the embedding process is the watermarked image which looks almost the same as the original cover image. Figure 3 shows the flowchart of the embedding process.

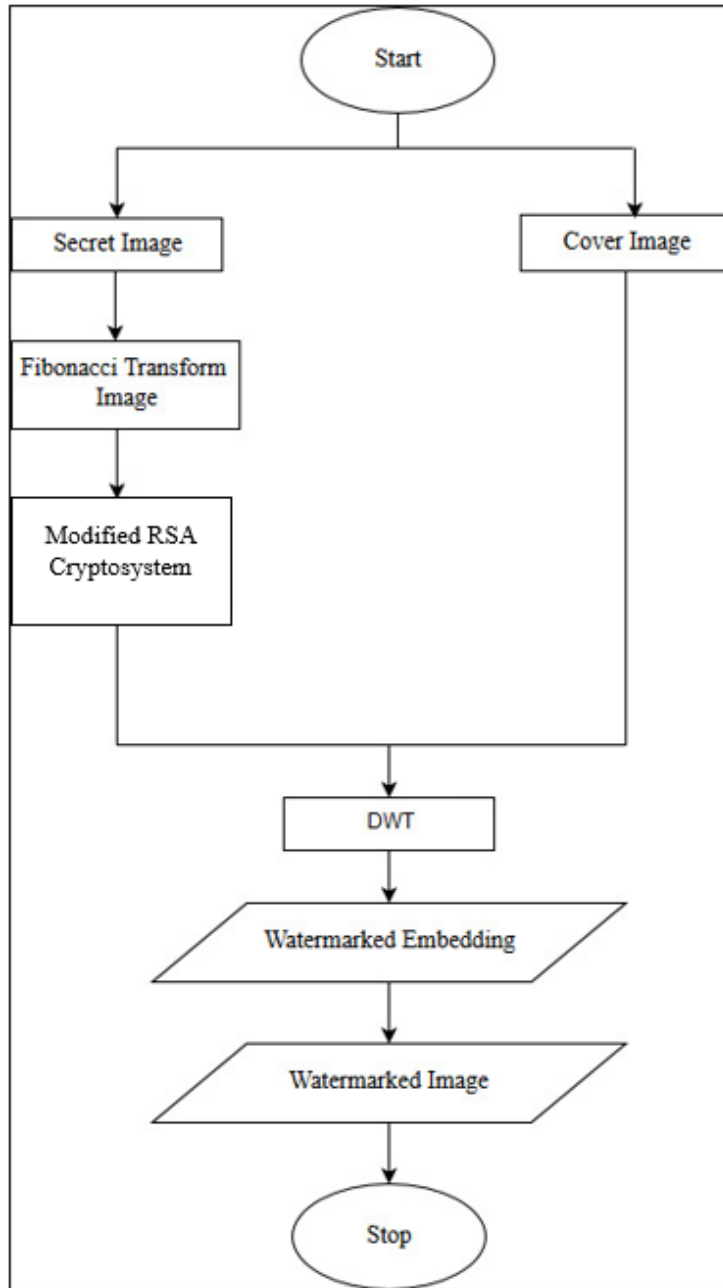


Fig. 3. Proposed embedding process flowchart.

First, the secret image is encrypted by Fibonacci transform and Modified RSA Cryptosystem. The cover image is then decomposed with DWT into four sub-bands LL, LH, HL, and HH. The LL sub-band is chosen as the location for watermark embedding. Finally, inverse DWT is performed to reconstruct the watermarked image. The detailed steps of the embedding process are shown in Algorithm 5.

Algorithm 5 Proposed Algorithm Embedding Process
<p>Input: Cover image A and secret image B Output: Watermarked image C</p> <ol style="list-style-type: none"> 1. Read the cover image is 512×512 pixels colour image and secret image is also 512×512 pixels colour image. 2. Carry out the Fibonacci number transform on the colour secret image to construct scrambled secret image. 3. Encrypt the scrambled secret image using Modified RSA Cryptosystem to form encrypted-scrambled secret image. 4. Make image carrier with a size of 8 by 8 blocks, transform every block by DWT 5. Embeds the encrypted-scrambled secret image into the transformed cover image to get the embedded image. 6. Apply the inverse DWT on the embedded image and construct the watermarked image. Select three large prime numbers p, q, r to form $R = nqr$.

The output of the embedding process is the watermarked image C . The private key d is saved for decryption in the extraction process.

3.2. Extraction process

In the extraction process, the secret image is removed from the watermarked image with the reverse of the embedding process. Any encryption on the secret image is also decrypted with the help of the private key previously used to encrypt it. By the end of the extraction, the outputs extract the cover image, and the secret image is obtained. The extracted images look like the original images most of the time. Still, often not an exact copy since the images go through processing or distortion attacks may have been applied to the watermarked image before the extraction process. Figure 4 shows the flowchart of the extraction process, and the steps are shown in Algorithm 6.

The watermarked image is first decomposed by DWT to extract the secret image from the LL sub-band. After the encrypted-scrambled secret image is extracted, the cover image can be reconstructed by inverse DWT to obtain the extracted cover image. The encrypted-scrambled image is then decrypted by Modified RSA Cryptosystem (Algorithm 3), then descrambled by Fibonacci transform (Eqn. 1).

The output of the extraction process is the possibly attacked extracted cover image and extracted secret image. The secret image needs to be extracted with the proper method to ensure the extracted secret image is undistorted and the message is properly sent.

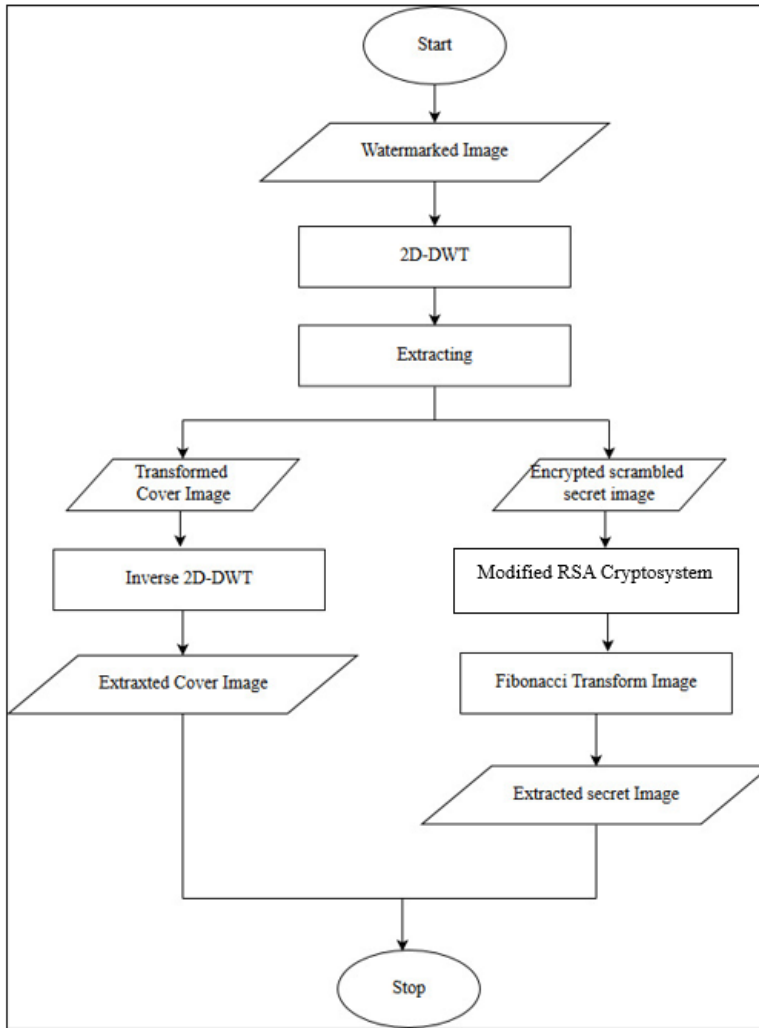


Fig. 4. Extraction process flowchart.

Algorithm 6 Proposed Algorithm Extraction Process
<p>Input: Watermarked image C</p> <p>Output: Extracted cover image A_{ext} and extracted secret image B_{ext}</p> <ol style="list-style-type: none"> 1. Input the colour watermarked image. 2. Apply back DWT into the watermarked image to get the embedded image. 3. Extract the embedded image to get the transformed cover image and encrypted-scrambled secret image. 4. Apply inverse DWT on transformed cover image to construct the extracted cover image. 5. Decrypt the encrypted-scrambled secret image using Modified RSA Cryptosystem and get the scrambled secret image. 6. Descramble the scrambled secret image using Fibonacci number Transform to obtain the extracted secret image.

3.3. Performance evaluation

The evaluation of the proposed algorithm involves calculating the peak signal-to-noise ratio (PSNR) between the watermarked image and the original cover image for imperceptibility and the structural similarity index (SI) value between the watermarked image and the original cover image for robustness.

3.3.1. Peak signal-to-noise ratio

The PSNR value between the watermarked image and the original cover image determines the imperceptibility of the watermark. Eq. (2) shows its calculation [14].

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

where f and g are the original image and watermarked image respectively, MSE is the cumulative mean squared error between the watermarked image and the original cover image. The MSE is calculated using Eq. (3) as follows [11].

$$MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2 \quad (3)$$

where M and N are the sizes of the digital image, and f_{ij} and g_{ij} is the pixel value at position (i, j) in the images f and g respectively. The value of the PSNR is lower when there is more noise. Therefore, a higher PSNR shows that the image is distorted less by the embedded watermark.

3.3.2. Structural similarity index value

The SI value represents the similarity between two images. The SI value between the watermarked image and original cover image is calculated. Eq. (4) shows the calculation method for SI values.

$$SI(x, y) = \frac{(2\mu_A\mu_B + c_1)(2\sigma_{AB} + c_2)}{(\mu_A^2 + \mu_B^2 + c_1)(\sigma_A^2 + \sigma_B^2 + c_2)} \quad (4)$$

where μ_A and μ_B are averages of the images A and B respectively, σ_A^2 and σ_B^2 are variances of images A and B respectively, σ_{AB} is the covariance of images A and B, and $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ are two variables to stabilize the division with the weak denominator, where L is the dynamic range of pixel values and $k_1 = 0.03$ and $k_2 = 0.01$.

4. Performance Evaluation or Computational Analysis

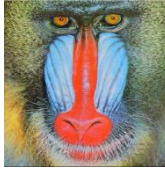



















The proposed algorithm is evaluated in this section. The watermarked images generated are shown, as well as the results after image processing attacks. The PSNR and SI values are also calculated. A graph will be presented to show the difference in performance between different cover images. Finally, comparisons with other similar algorithms will be made.

4.1. Generated images

Images may be distorted enough by intentional or unintentional attacks that the secret images become unusable, so it need to be robust to prevent this occurrence.

Table 1 shows the watermarked images of Baboon, Lena, Sunflower, and Building without or when under four different attacks separately.

Table 1. Watermarked images before and after the attack.

Without Attack	Gaussian Noise Attack	Contrast Attack	Blurring Attack	Sharpening Attack
				
				
				
				

4.2. Performance evaluation

The results of the PSNR and SI value calculations are shown in this section, then compared with other watermarking algorithms.

4.2.1. Peak signal-to-noise ratio

The PSNR value between the watermarked image and the original cover image is compared. Table 2 shows the results.

Based on Table 2, the PSNR value significantly drops when the watermarked image is under attack. The contrast attack affects the PSNR the most, as can also be subjectively observed in Table 1. The PSNR is least affected by the sharpening attack. Figure 5 is presented to aid the comparison between cover images.

Table 2. PSNR value evaluation.

Cover Image	PSNR value				
	Without Attack	Gaussian Noise Attack	Contrast Attack	Blurring Attack	Sharpening Attack
Baboon	59.5563	20.1614	15.1793	20.8595	25.5371
Lena	59.0098	19.7832	18.0031	29.0031	36.415
Sunflower	59.4844	20.7372	17.4073	29.7873	35.3561
Building	60.0065	20.5038	15.7318	22.743	29.4005

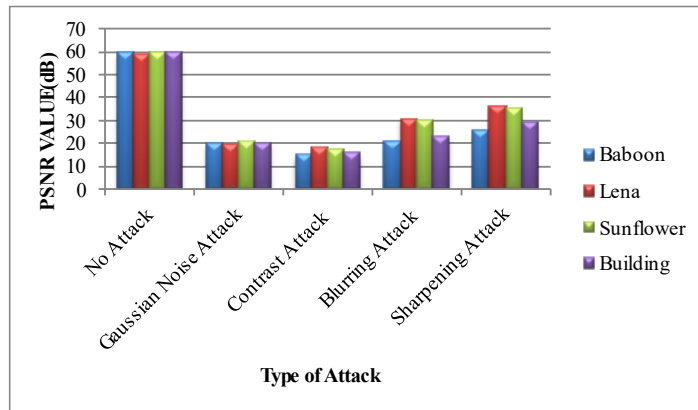


Fig. 5. PSNR value bar graph.

All the images show a similar value without any attacks. Notably, the Lena and Sunflower images are less affected by the attack than the Baboon and Building images, except for the gaussian noise attack where the Lena image is most affected. Therefore, the choice of cover image is also important for better results. Table 3 shows the comparison with other algorithms.

Table 3. PSNR comparisons.

Cover Image	PSNR value (dB)										Proposed method
	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]	[24]	
Lena	42.7869	-	73.12	62.64	43.29	44.36	101.97	58.36	51.8848	46.0999	59.5563
Baboon	37.3475	-	71.48	-	-	-	-	57.39	47.4997	46.0817	59.0098
Sunflower	-	66.431	-	-	-	-	-	-	-	46.0804	59.4844

The results from the watermarked image not under attack are taken. Overall, the proposed algorithm seems to be more imperceptible than six out of ten of the compared watermarking algorithms, based on the tested images. The results of the proposed algorithm do not deviate too far from the other algorithms and are higher than some of them, so the proposed algorithm is competent.

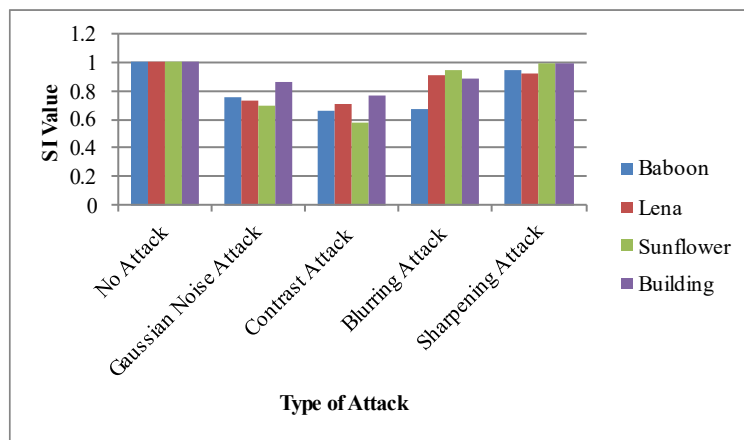
4.2.2. Structural similarity index value

The SI value is computed between the watermarked image and the original cover image. Table 4 shows the results of the calculation.

Table 4. SI value evaluation.

Cover Image	SI value				
	Without Attack	Gaussian Noise Attack	Contrast Attack	Blurring Attack	Sharpening Attack
Baboon	1.0000	0.7538	0.6582	0.6673	0.9464
Lena	1.0000	0.7311	0.7123	0.9133	0.9233
Sunflower	1.0000	0.7011	0.5792	0.94322	0.9888
Building	1.0000	0.8579	0.7693	0.89166	0.9889

Note that the values are rounded to the nearest four decimal places. Without any attacks, the SI value is near perfect, meaning that the watermarked image is very similar to the original cover image. Consistent with the PSNR results from Table 2, the SI value in Table 4 also shows that the contrast attack affects the watermarked image the most, while the sharpening attack affects it the least. Figure 6 shows the graph to compare the SI values between the different cover images.

**Fig. 6. SI value bar graph.**

Based on Fig. 6, the building cover image much higher SI value than the other images for the gaussian noise attack and contrast attack and is still highest in the sharpening attack. The Sunflower image also seems to perform much better against blurring and sharpening attack. Lastly, the Baboon image seems to be especially weak against the blurring attack.

5. Conclusions

This section concludes with the results and discussions of the proposed algorithm. The resulting watermarked image is very similar to the original cover image with a PSNR value of 59-60dB. The watermarked image seems to be affected most by the contrast attack, and least affected by the sharpening attack. The choice of the cover image is also important to achieve better results. The algorithm is also quite competent, based on the comparisons previously made. The cryptosystem applied in the algorithm is applicable and suitable for digital image watermarking. Although it was designed with the encryption of words or numbers in mind, it has been proven that images can also be encrypted by the cryptosystem.

For future works, several plans involving the algorithm could be carried out. The algorithm can successfully embed the secret image on a cover image, but its flexibility could be further improved by expanding the application to audio and video media. Alternative scrambling methods other than the Fibonacci transform scrambling can also be experimented on to decide on the best one to use or the most suitable based on the intended application of the watermarking algorithm. To further promote the use of cryptosystems in picture watermarking methods, exploring the potential integration of alternative cryptosystems, such as elliptic curves or blowfish cryptosystems, would be beneficial.

Abbreviations

DWT	discrete wavelet transform
PSNR	Peak signal-to-noise ratio
SSIM	Structural similarity index measure

References

1. Cox, I.J.; Miller, M.L.; Bloom, J.A.; Fridrich, J.; and Kalker, T. (2007). *Digital watermarking and steganography*. (2nd ed.). Elsevier.
2. Abdullatif, M.; Zeki, A.M.; Chebil, J.; and Gunawan, T.S. (2013). Properties of digital image watermarking. *Proceedings of the 2013 IEEE 9th International Colloquium on Signal Processing and its Applications*, Kuala Lumpur, Malaysia, 235-240.
3. Mahto, D.K.; and Singh, A.K. (2021). A survey of color image watermarking: state-of-the-art and research directions. *Computers & Electrical Engineering*, 93, 107255.
4. Verma, V.S.; and Jha, R.K. (2015). An overview of robust digital image watermarking. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, 32(6), 479-496.
5. Bhatt, K.; Singh, K.; and Saxena, A. (2023). Robust DWT-SVD watermarking algorithm with additional digital signature security for color images. *AIP Conference Proceedings*, 2724(1), 040003.
6. Liang, R.; Qin, Y.; Zhang, C.; Lai, J.; Liu, M.; and Chen, M. (2019). An improved Arnold image scrambling algorithm. *IOP Conference Series: Materials Science and Engineering*, 677(4), 042020.
7. Yang, L.; Shanyu, T.; Ran, L.; Liping, Z.; and Zhai, M. (2018). Secure and robust digital image watermarking scheme using logistic and RSA encryption. *Expert Systems with Applications*, 97, 95-105.
8. Ye, G.; Jiao, K.; Wu, H.; Pan, C.; and Huang, X. (2020). An asymmetric image encryption algorithm based on a fractional-order chaotic system and the RSA public-key cryptosystem. *International Journal of Bifurcation and Chaos*, 30(15), 2050233.
9. Razak, M.K.A.; Abdullah, K.; and Abd Halim, S.A. (2022). Non-blind image watermarking algorithm based on non-separable Haar Wavelet transform against image processing and geometric attacks. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 29(2), 251-267.
10. Razak, M.K.A.; Abdullah, K.; and Halim, S.A. (2022). Robustness of modified non-separable Haar wavelet transform and singular value decomposition for

- non-blind digital image watermarking. *Malaysian Journal of Mathematical Sciences*, 16(2), 289-316.
11. Alsabti, K.D.M.; and Hashim, H.R. (2016). A new approach for image encryption in the modified RSA cryptosystem using MATLAB. *Global Journal of Pure and Applied Mathematics*, 12(4), 3631-3640.
 12. Shanthi, P.; and Bhunaveswaran, R.S. (2016). Image watermarking using fibonacci transform. *Asian Journal of Information Technology*, 15(9), 1431-1436.
 13. Liu, J.; Huang, J.; Luo, Y.; Cao, L.; Yang, S.; Wei, D.; and Zhou, R. (2019). An optimized image watermarking method based on HD and SVD in DWT domain. *IEEE Access*, 7, 80849-80860.
 14. Vaish, A.; and Kumar, M. (2017). Color image encryption using MSVD, DWT and Arnold transform in fractional Fourier domain. *Optik*, 145, 273-283.
 15. Escalante-Ramirez, B.; and Gomez-Coronel, S. (2018). A perceptive approach to digital image watermarking using a brightness model and the Hermite transform. *Mathematical Problems in Engineering*, Volume 2018, Article ID 546363.
 16. Perwej, Y.; Parwej, F.; and Perwej, A. (2012). Copyright protection of digital images using robust watermarking based on joint DLT and DWT. *International Journal of Scientific & Engineering Research*, 3(6), 1-9.
 17. Mohamed, M.A.; Aboutaleb, H.M.A.A.M.; Abdel-Fattah, M.G.; and Samrah, A.S. (2015). Hybrid watermarking scheme for copyright protection using chaotic maps cryptography. *International Journal of Computer Applications*, 126(4), 13-26.
 18. Aparna, J.R.; and Ayyappan, S. (2015). Image watermarking using Diffie Hellman key exchange algorithm. *Procedia Computer Science*, 46, 1684-1691.
 19. Rao, S. M.; Srujan, R.; Madhukar, V.; Rahul, H.S.; and Sandeep, K.V. (2016). A secure color image watermarking scheme using RSA encryption. 6(5), 4948-4950.
 20. Chang, Y.-F.; and Tai, W.-L. (2013). A block-based watermarking scheme for image tamper detection and self-recovery. *Opto-Electronics Review*, 21(2), 182-190.
 21. Zhou, X., Zhang, H.; and Wang, C. (2018). A robust image watermarking technique based on DWT, APDCBT, and SVD. *Symmetry*, 10(3), 77.
 22. Zhang, Z.; Wu, L.; Yan, Y.; Xiao, S.; and Sun, H. (2017). An improved reversible image watermarking algorithm based on difference expansion. *International Journal of Distributed Sensor Networks*, 13(1), 1550147716686577.
 23. Dhar, P.K.; Hasan, R.; and Shimamura, T. (2018). Color image watermarking based on radon transform and Jordan decomposition. *Digital Image and Video Watermarking and Steganography*. IntechOpen.
 24. Meenpal, T. (2018). DWT-based blind and robust watermarking using SPIHT algorithm with applications in tele-medicine. *Sādhanā*, 43(1), 1-12.