Malaysian Journal of Mathematical Sciences 19(3): 857–870 (2025)
https://doi.org/10.47836/mjms.19.3.05

Malaysian Journal of Mathematical Sciences

Journal homepage: https://mjms.upm.edu.my

LSB Technique for Image Steganography Based on RC4 Algorithm and ECDHKE Protocol

Al Saedi, S. A. H. [1] and Al Saffar, N. F. H. * [1]

[1]*Department of Mathematics, Faculty of Computer Science and Mathematics, University of Kufa, Iraq*

E-mail: najlaa.hameed@uokufa.edu.iq
**Corresponding author*

Received: 4 December 2024
Accepted: 6 March 2025

Abstract

Image-based steganography using the least significant bits technique makes detecting hidden plaintext inside an image file difficult. This method hides not only the message itself but also the fact that it is being sent. This work used several tools to build a proposed algorithm to increase hidden security information, where the keys will be generated using the Diffie-Hellman technique that depends on points of an elliptic curve. At the encryption level, the RC4 (Rivest Cipher 4) technique will be used to encrypt the plaintext. The last level will use the least significant bit technique within an image. The tests (histogram analysis, mean square error, peak signal-to-noise ratio) conducted on the newly developed algorithm revealed something quite intriguing; it is not just straightforward and efficient but boasts a remarkably high level of security for information transfer. These results were supported by calculating the execution time of the proposed algorithm and comparing it with previous algorithms.

Keywords: least significant bits technique; image-based steganography; Diffie-Hellman technique; elliptic curve; RC4 technique.

1 Introduction

Steganography is the science and art of hiding data or messages in other media so that the existence of the secret message cannot be known. In general, the steganography technique inserts information piece by piece into a medium, so that the information appears less dominant than the protective media [20]. When it comes to messages, data, or information, security and confidentiality are paramount. The truth and authenticity of that information are crucial, not just when it's dispatched but also when it finally lands in someone else's hands. Indeed, when messages, data, or any kind of information get sent out into the world, they can quickly lose their value if they fall into the wrong hands [26].

Cryptography, often described as the intricate art of encoding and decoding messages, has evolved far beyond its basic definition in the Concise Oxford Dictionary from 2006. While it initially conjured images of clandestine communication and mysterious ciphers, today's landscape paints a much broader picture. It intertwines with digital signatures and the complex protocols that ensure that the secret keys don't fall into the wrong hands. This field has woven itself into the very fabric of numerous scientific disciplines, becoming not just a speciality but a pivotal cornerstone in a world driven by information and security. The multifaceted nature of cryptography now sparks curiosity and innovation across various spheres, making it a captivating area of exploration and study [24].

Steganography and cryptography both share a common goal: shielding sensitive information from prying eyes. Each with its own unique flair, yet neither is infallible [4]. While both methods can effectively obscure data, they each have vulnerabilities that can be exploited. This is precisely why savvy security experts advocate for a dual approach, layering these techniques to create a robust fortification against potential intruders [12]. Indeed, cryptography is primarily concerned with concealing the message's content, while steganography's mission is to hide the very existence of the message itself. Steganography, often seen as an intricate art of concealment, meticulously disguises information in ways that render it confidential. It's like tucking a treasure map within a mundane photograph, ensuring that anyone who happens upon it remains blissfully unaware of its significance. Together, these techniques create a tapestry of security, weaving complexity into the very fabric of digital communication [21].

Steganography is a technique used to conceal sensitive information by using the cover object as an image. Text steganography, image steganography, audio steganography, video steganography, and protocol steganography are the five main categories of steganography techniques [19]. As mentioned, image steganography is one of the types; it uses images to conceal dispatches. The term "cover image" refers to the original image before any information is concealed inside it. It is known as the stego image when the information has been concealed within it. These images must be the same for the human eye [25].

Steganography aims to prevent detection by employing methods of concealing information in a regular, non-secret file. The most common images used for steganography are cover images, and the information that needs to be concealed throughout the algorithm's construction will be text. In this work, a combination of the Diffie-Hellman technique that depends on points of an elliptic curve and the RC4 technique will be used to generate the keys and encrypt the plain text, and then the least significant bits technique will be the tool for hiding the encrypted text within the selected image.

This work is organized as follows: Section 2 will be a review of the literature, which will give a brief overview of image steganography and all concepts that will be used as a tool for constructing

the proposed algorithm (Least Significant Bits (LSB) technique, RC4 algorithm, and Elliptic Curve Diffie-Hellman Key Exchange). Section 3 will be about the proposed algorithm, followed by the security analysis in Section 4, where the proposed algorithm will be tested using a histogram, mean square error, and peak signal-to-noise ratio. The time running of all levels of the proposed algorithm will be discussed in the time implementation, Section 5. Section 6 will be a conclusion and will show how the other researchers can develop the total idea of this work.

2 Review of Literature

2.1 Image steganography

As an art of information concealment, steganography deals with methods of conveying and concealing data while maintaining confidentiality [20]. As mentioned, using the cover object as an image is a steganography method of some sort [21]. Steganography techniques: text, image, audio, video, and protocol are the five main categories of steganography techniques. The second kind involves hiding any text in an image to guarantee its security. In today's digital world, text security has grown increasingly crucial as a result of the rapid development of the internet [1]. One of the most common techniques used to hide text within an image is to replace each bit of text with the least significant bit within the image; this technique is called the Least Significant Bits (LSB) technique [6]. A brief overview of this technique will be presented in the next subsection.

2.2 Least Significant Bits (LSB) technique

The most popular method of data concealing is called LSB, in which the data that needs to be hidden is replaced with the least important portions of the image pixels. Because the changing of the LSB of the image does not result in significant changes to the image, the image obtained after embedding is essentially homologous to the original image [13]. Two types of this technique (1-LSB and 2-LSB) have been discussed in [11] through a combination of them to find the difference between them and to clarify the positive and negative aspects of each of them; the elliptic curve cryptography was involved; this work is based on the first type. Figure 1 shows how the LSB is determined for 8 bits.

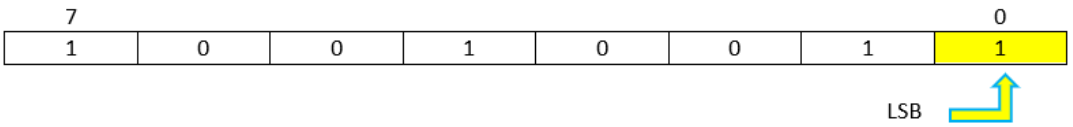


Figure 1: LSB value of 201 that has 8 bits.

In terms of numerical illustration, assuming that the text $A = (65)_{10} = (01000001)_2$ needs to be hidden within the selected image with 3 rows as:

Row 1: 11111000 11001100 10101011 = 248 204 171

Row 2: 10101010 11001001 00000011 = 170 201 3

Row 3: 01101101 11001010 00110011 = 109 202 51

The result after replacing the bits of the letter A with the least significant bits in the selected image is:

Pixel 1: 11111000 11001101 10101010 = 248 205 170

Pixel 2: 10101010 11001000 00000010 = 170 200 2

Pixel 3: 01101100 11001010 00110011 = 108 202 51

It is clear that the change if it occurs after the replacement process (increase or decrease by one) is considered slight, which means that the colour tone of the image at that pixel will not be clear to the human eye. This technique will be a tool to hide the encrypted text within selected images in this work.

2.3 RC4 algorithm

As a stream cipher, RC4 is a symmetric key technique that handles units or data input concurrently. Usually, a byte, or occasionally a bit serves as the data unit. This algorithm does not need to wait for a specific amount of input data before processing or adding more bytes for encryption if it is used in any way. Ronald Rivest created this algorithm in 1984, and it was made publicly available on mail and news groups in 1994 using pseudonyms [14]. This algorithm's encryption process was split into the following two sections:

1. An evaluation of the RC4's initialization that focuses on the initialization of the key scheduling algorithms (KSA).
2. An analysis of the keystream-generating output, focusing on the internal state and round-running procedure of the pseudo-random generation algorithm (PRGA).

Random numbers are crucial to cryptography procedures. Cryptographic objects that involve several unpredictable components include keys, nonces, block padding initialization vectors, and obstacles [22]. As mentioned, the pseudorandom number generator and key scheduling are the two primary parts of the RC4. Using an initial permutation, a (random) key K of l-byte length, and two pointers i and j, the key scheduling creates an internal random permutation S of values ranging from 0 to 255. The longest possible key length is $l = 256$ bytes [17]. There are four steps to do the encryption using the RC4 algorithm:

First Step: Initialization of S.
The preliminary operations can be summarized as follows:
For $i = 0$ to 255,
$S[i] = i$,
$T[i] = K[i] \mod keylen$,
where $keylen$ refers to the length of the given key.

Second Step: Initial permutation of S.
The preliminary operations can be summarized as follows:
$J = 0$,
For $i = 0$ to 255,
$j = (j + S[i] + T[i]) \mod 256$,
$swap(S[i], S[j])$,

where the $swap$ function is used to swap the values of two variables; indeed, this function is commonly used when the positions of two values need to change between two variables without losing data.

Third Step: Key generation.
The preliminary operations can be summarized as follows:
$i, j = 0$,
if it is true,
then $i = (i + 1) \mod 256$,
$j = (j + S[i]) \mod 256$,
$swap(S[i], S[j])$,
$T = (S[i] + S[j]) \mod 256$,
$K = S[t]$.

Fourth Step: Encryption.
To encrypt the plaintext, XOR ing the value of K with the next byte of the plaintext.

The decryption level of the RC4 algorithm is achieved by XOR ing the value of K with the next byte of the ciphertext. Figure 2 shows how the RC4 algorithm works to generate its keys.

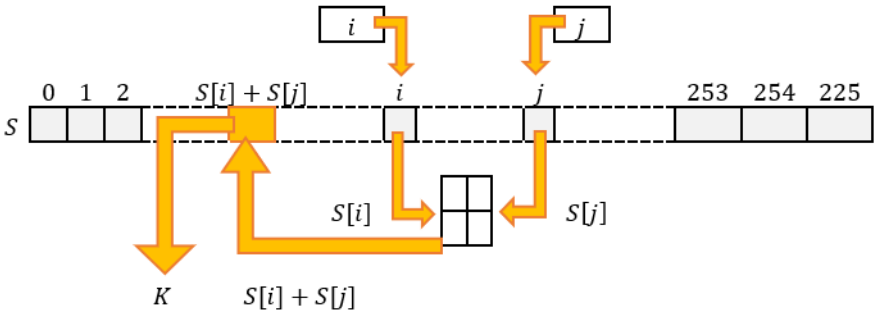


Figure 2: Generating keys with RC4 algorithm.

In terms of numerical illustration, assuming that there are 4 bytes, $K[0] = 2$, $K[1] = 5$ to encrypt the plaintext "**HI**" with **H**$= 01001000$ and **I** $= 01001001$.

First Step: Initialization of S:
For $i = 0$ to 3,
$S[i] = K = 0123$,
$T[i] = K[i] \mod keylen = 2525$.

Second Step: Initial permutation of S:
Iteration 1: $i, j = 0$,
$j = (j + S[i] + T[i]) \mod 4 = (0 + 0 + 2) \mod 4 = 2$,
$Swap(S[i], S[j]) = Swap(S[0], S[2]) = \boxed{2103}$.
Iteration 2: $i = 1, j = 2$,
$j = (j + S[i] + T[i]) \mod 4 = (2 + 1 + 5) \mod 4 = 0$,
$Swap(S[1], S[0]) = \boxed{1203}$.
Iteration 3: $i = 2, j = 0$,
$j = (j + S[i] + T[i]) \mod 4 = (0 + 0 + 2) \mod 4 = 2$,
$Swap(S[2], S[2]) = \boxed{1203}$.

Iteration 4: $i = 3, j = 2$,
$j = (j + S[i] + T[i]) \mod 4 = (2 + 3 + 5) \mod 4 = 2$,
$Swap(S[3], S[2]) = \boxed{1230}$.

Third Step: Key generation:
For the letter **H**:
$i = 0, j = 0$,
$i = (i + 1) \mod 4 = (0 + 1) \mod 4 = 1$,
$j = (j + S[i]) \mod 4 = (0 + 2) \mod 4 = 2$,
$Swap(S[1], S[2]) = 1320$,
$T = (S[i] + S[j]) \mod 4 = (3 + 2) \mod 4 = 1$,
$K_0 = S[T] = S[1] = \boxed{3}$.
For the letter **I**:
$i = 1, j = 2$,
$i = (i + 1) \mod 4 = (1 + 1) \mod 4 = 2$,
$j = (j + S[i]) \mod 4 = (2 + 2) \mod 4 = 0$,
$Swap(S[2], S[0]) = 2310$,
$T = (S[i] + S[j]) \mod 4 = (1 + 2) \mod 4 = 3$,
$K_1 = S[T] = S[3] = \boxed{0}$.

Forth Step: Encryption:
For the letter **H**: $01001000 \oplus 00000011 = 01001011$,
For the letter **I**: $01001001 \oplus 00000000 = 01001001$,
So, the ciphertext is 0100101101001001 or "**KI**".
This algorithm will be a tool to encrypt the plaintext to ciphertext in this work.

2.4 Elliptic Curve Diffie Hellman Key Exchange (ECDHKE)

Elliptic curves which are a kind of Diophantine equation [9] have been extensively studied by number theorists for more than hundreds of years, only for their mathematical beauty, not for their applications. However, in the late 1980s and early 1990s, many important applications of elliptic curves in both mathematics and computer science were discovered [8]; the applications of elliptic curves in cryptography were not found until the middle of 1980 when Miller [18] and Koblitz [16] introduced their works, where the elliptic curve was a tool for cryptography; since these works, the concept of elliptic curve cryptosystem (ECC) was invented. ECC offers a high level of security with smaller key sizes, making it ideal for applications that run on small devices that have power and memory; indeed, it is a good tool for combining encryption with steganography [10]. The first protocol in the public key cryptosystem was the Diffie-Hellman key exchange protocol [7], which was put forth by Diffie and Hellman in 1976. The elliptic curve counterpart of the Diffie-Hellman key exchange is called ECDHKE. There are no messages involved in this system; it is only a way to exchange keys. Assume for the moment that $Alice$ and Bob are two users who wish to utilize the ECDHKE protocol to exchange their keys. The following is the ECDHKE procedure:

- An elliptic curve $E : y^2 \equiv (x^3 + ax + b) \mod p$ over a prime field F_p with a base point B of order n is agreed upon by $Alice$ and Bob.
- $Alice$ (Bob) chooses a random and secret $e(d)$ such that $1 \leqslant e, d \leqslant n - 1$ and computes $eB(dB)$ and sends it to Bob ($Alice$).

  Then, eB and dB are public, and e and d are secret.
- $Alice$ (Bob) computes the secret key edB (deB).

After these steps, $Alice$ and Bob will have the same point, which belongs to points of the elliptic curve E over F_p. This protocol's main operation and all protocols are based on elliptic curve scalar multiplication [3]. To provide a numerical example, $Alice$ and Bob will use the ECDHKE protocol to exchange their keys in the manner described below:

- $Alice$ and Bob agree upon an elliptic curve $E : y^2 \equiv (x^3 + x + 4) \mod 23$ over a prime field F_{23} and a base point $B = (7, 3)$ of order 29.
- $Alice$ (Bob) chooses a random and secret $12(23)$ such that $1 \leqslant 12, 23 \leqslant 28$ and computes $12(7, 3) = (13, 11)$,$(23(7, 3) = (0, 2))$ and sends it to Bob $(Alice)$.

  Then, $(13, 11)$ and $(0, 2)$ are public and 12 and 23 are secret.
- $Alice$ (Bob) computes the secret key $12(0, 2) = (17, 9)$ $(23(13, 11) = (17, 9))$.

That is the shared key $(17, 9)$. This protocol will be a tool to generate a key for the RC4 algorithm in this work.

3 Proposed Algorithm

Cryptography and steganography are two concepts that will be involved in this work, where the purpose of using them is to shield information from unauthorized access. Steganography and cryptography are different in that steganography focuses on keeping a message's existence hidden, while cryptography focuses on keeping a message's contents secret. Combining the cryptosystem with steganography techniques is a novel way to secure information transport. In this work, this combination of the steganography technique (LSB) and the cryptosystem protocols ECDHKE and RC4 will be the main topic. The proposed algorithm will deal with this issue because the security of any encryption technique rests on how to stop hackers from accessing confidential data. If two users ($Alice$ and Bob) are considered to be two users for implementing the proposed algorithm to cipher the plaintext M and hide it within a selected image, they have to follow the following steps:

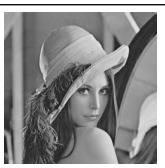
- $Alice$ and Bob will choose an elliptic curve $E : y^2 \equiv (x^3 + ax + b) \mod p$ over a prime field F_p and a base point B of order n. (All these parameters will be public keys for this algorithm.)
- They have to implement the ECDHKE protocol to generate the shared key $sh_K = (k_1, k_2)$.
- They have to compute the $RC4_K = (k_1)^{k_2}$.
- $Alice$ (the sender) will implement the RC4 algorithm to encrypt the plaintext M to ciphertext C using the key $RC4_K$.
- For steganography level, $Alice$ will do the following steps:
  - Convert the selected image to a grey image.
  - Convert the result to binary form; in this binary there are $8-$bits for every row.
  - Convert the ciphertext C to binary form; the size of the new form (say $h \times l$); h and l will be another public keys.
  - Replace the bits of C with the least significant bits in the selected image.
  - Convert the output to $Stegimage$.

- $Alice$ will send $Stegimage$ to Bob within a public channel.
- Bob (the recipient) will convert $Stegimage$ to binary form; in this binary there are $8-$bits for every row.
- He will use the public keys h and l to implement the LSB to determine the ciphertext C.
- He will implement the decryption step of RC4 to convert C to M using $RC4_K$.

4 Security Analysis

By computing the histogram, mean square error (MSE), and peak signal-to-noise ratio (PSNR), the security performance of the proposed technique is examined. The proposed algorithm's resilience to statistical attacks can next be assessed, where different plain texts were employed to evaluate the proposed algorithm's capacity for embedding. The selected images that were used to test the proposed algorithm are shown in Table 1. All grayscale images with different sizes that were selected from the standard database of $MATLAB\ R2024a.$ with $E : y^2 \equiv (x^3 + x + 4) \mod 23$ over a prime field F_{23} and a base point $P = (7, 3)$ of order 29. and three different plaintexts with different sizes. Utilizing Matlab $R2024a\ (24.1.0.2537033)\ 64-$bit software on a workstation equipped with $Intel\ ®\ Core\ ™\ i7\ CPU\ (2.70\ GHz)$, $64\,GB\,RAM$, and $Microsoft\ Windows\ 11\ Pro$ as the operating system was used to perform all the tests in this document, including RC4, ECDHKE, and steganography processing that uses specific images to conceal information.

Table 1: The selected images that were utilized to assess the proposed algorithm.

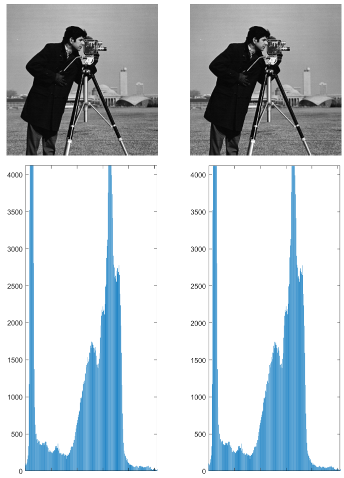
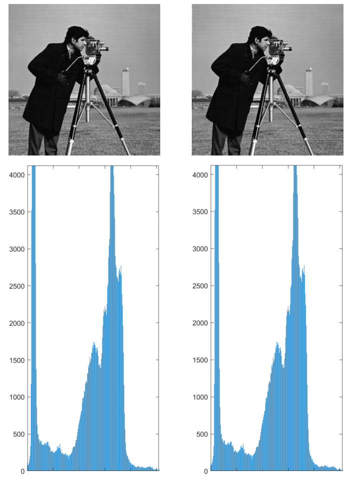
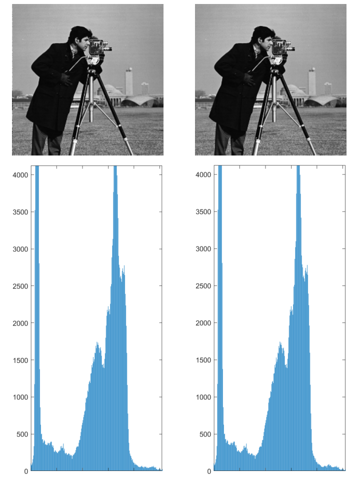
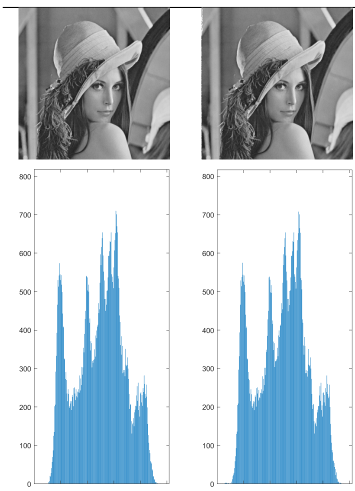
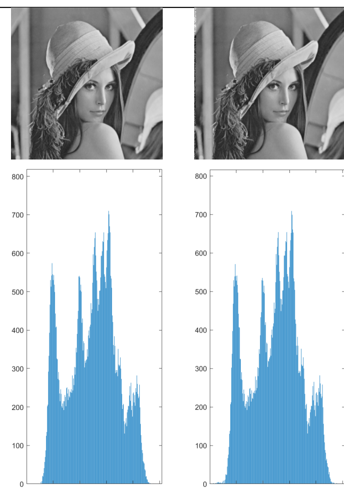
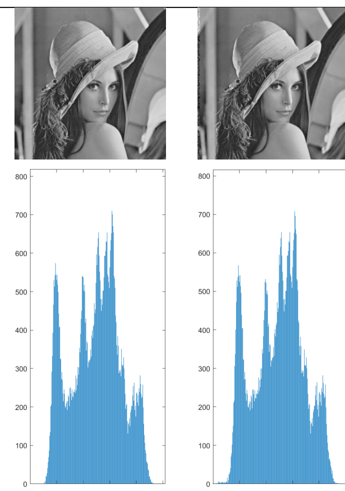
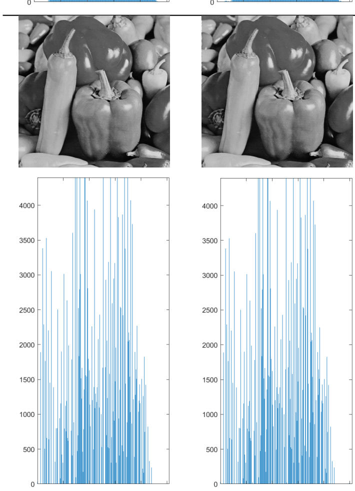
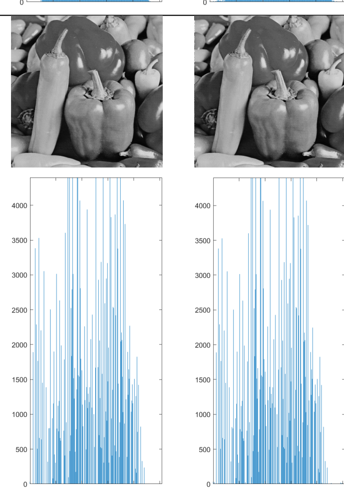
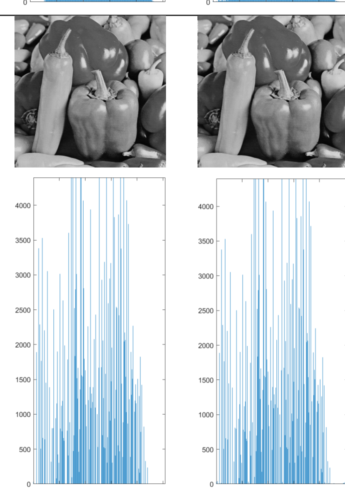
First Image	Second Image	Third Image
		

4.1 Histogram analysis

A key component of effective photo steganography is that the little colour variations between the original and the images that had ciphering plaintext are almost imperceptible to the human eye. The efficacy of any proposed algorithm is demonstrated by its capacity to hide the modifications. The frequency of pixel intensities in a particular image is shown by the histogram [2], For the proposed algorithm, Table 2 displays the histograms of the original selected images and the images that had ciphering plaintext.

It is evident that the histogram of a stego image is nearly identical to the histogram of the cover image, making it extremely difficult to analyze encrypted texts that have been concealed. The efficiency of this proposed approach is shown by the conducted trials and the separate histograms created for the cover and stego images. Even with the maximum amount of secret plaintext input, the histograms of the original image and the stego image are almost identical, as seen in Table 2. High security can be ensured to withstand statistical assaults, and no valuable information can be recovered from the stego image as a consequence.

Table 2: Histograms of the original image and image had ciphertext.

First Plaintext	Second Plaintext	Third Plaintext
		
		
		

4.2 Mean square error and peak signal-to-noise ratio

The stego image's quality will be assessed in relation to the cover image using the mean square error (MSE) and peak signal-to-noise ratio (PSNR), which are effective metrics for gauging the invisibility of embedded data inside an image. The results of the proposed approach applied to three selected images are shown in Table 3. The findings of the first experiment, which employed the first plaintext (containing 37 bits), indicate that the MSE and PSNR differed across the three images; the first image's MSE value was precise 0, but the PSNR value was greater. The second image has a high PSNR value of 53.1454 and an MSE value of 0.3152. In the same way, the third image's PSNR was 58.8060 and its MSE was 0.0856.

Table 3: PSNR and MSE values of the original image and the image that had the ciphertext of the first plaintext.

Images	Tests	
	MSE	PSNR
First Image	0	inf
Second Image	0.3152	53.1454
Third Image	0.0856	58.8060

Table 4 presents the results of the second plaintext (with 76 bits) on the three selected images. Regarding the first image, the MSE was 0.0788, and the PSNR was 59.1660. For the second image, the MSE was 0.6537, and the PSNR was 53.1454. For the third image, the MSE was 0.1566, and the PSNR was 56.1825. The MSE and PSNR values for each of the three images changed, according to the findings. A higher PSNR value suggests that the stego image's quality is higher, while a lower MSE value indicates that there is almost no difference between the original image and the stego image that had ciphertext.

Table 4: PSNR and MSE values of the original image and the image that had the ciphertext of the second plaintext.

Images	Tests	
	MSE	PSNR
First Image	0.0788	59.1660
Second Image	0.6537	53.1454
Third Image	0.1566	56.1825

The third plaintext's MSE and PSNR results for the three chosen images are shown in Table 5. According to the findings, the first image had the highest PSNR and the lowest MSE, whereas the second image had the lowest PSNR and the highest MSE. This suggests that, in comparison to the other two images, the first image had the best image quality and the least amount of distortion degradation following the application of the steganographic method. These findings can be utilized to enhance steganographic algorithms in the future and offer insights into how well the algorithm performs for various images.

Table 5: PSNR and MSE values of the original image and the image that had the ciphertext of the third plaintext.

Images	Tests	
	MSE	PSNR
First Image	0.1060	57.8765
Second Image	1.0389	47.9651
Third Image	0.2228	54.6524

Overall, it is obvious that the values of MSE and the PSNR are dependent; if the first values drop, the second values rise, which means the quality of the images that had ciphering plaintext will improve as the PSNR value increases.

In terms of comparing the proposed algorithm with previous related works, a recently published article [12] that used the LSB technique as a tool for information hiding was chosen to compare the results of the proposed algorithm with the results obtained by this article. The value for the PSNR for the first image was 42.75 and the MSE was 3.47, whereas the outcomes demonstrate that the suggested approach yields the highest PSNR values (inf, 59.1660 and 57.8765) and the lowest values of MSE (0, 0.0788 and 0.1060) for the first image with three plaintexts.

5 Time Implementation

An essential step is figuring out how long the proposed algorithm will take to implement. Algorithms will benefit from using a lot of data with a faster algorithm. A high-performance algorithm must therefore take implementation time and security into account. The results of applying the suggested technique to three chosen photographs are displayed in Table 6. The implementation time using the proposed algorithms in [27] was 2.52 seconds, while the implementation time using the proposed algorithms in [28] and [15] was 2.018 seconds and 1.642 for the only recipient procedure, respectively. Table 6 shows that a proposed algorithm can encrypt the plaintexts and hide the ciphertexts within images (as a sender procedure) and then, retrieve the ciphertexts from the images and decrypt them to the plaintexts (as a recipient procedure) so quickly. Indeed, the combined duration of the two processes is less than 0.06 seconds.

Table 6: Implementation time of the proposed algorithm on the selected images.

Images	Execution time in seconds:		
	as a sender procedure	as a recipient procedure	Total
First Image	0.0150	0.0133	0.0283
Second Image	0.02150	0.0222	0.0437
Third Image	0.0349	0.0187	0.0536

6 Conclusion and Future Work

One of the most popular image method attacks is the brute force attack. It entails closely observing every potential stego key until the right one is identified. In the proposed algorithm, the key was generated using a secure algorithm to exchange keys (ECDHKE), where its security is based on a hard mathematical problem (elliptic curve discrete logarithm problem); indeed, this problem needs a fully exponential time to solve. In terms of the plaintext size, the attacker should be aware of this part; in the proposed algorithm, the RC4 technique with a stronger key can serve any plaintext size. This work introduced a proposed algorithm to hide information using the LSB technique, which uses the ECDHKE and RC4 techniques to conceal plaintext within an image.

Furthermore, by converting the image to binary form, the proposed approach improves its overall security and makes it more resilient to attacks. All in all, a secure algorithm to hide text within an image based on the ECDHKE and RC4 techniques has been proposed and implemented. The results of histogram analysis, time implementation, and the values of MES and PSNR proved to demonstrate the algorithm's efficiency. Future studies could explore extending this procedure to use audio or video files as cover media. Additionally, Blowfish [23] encryption or its modification [5] could be integrated to enhance the encryption process before embedding the data into the cover file.

Acknowledgement The authors would like to sincerely thank everyone whose advice, encouragement, and contributions were crucial to the planning and accomplishment of this paper.

Conflicts of Interest The authors declare that there are no conflicts of interest.

References

[1] E. A. Abbood, R. M. Neamah & S. Abdulkadhm (2018). Text in image hiding using developed LSB and random method. *International Journal of Electrical & Computer Engineering*, *8*(4), 2091–2097. https://doi.org/10.11591/ijece.v8i4.pp2091-2097.

[2] N. Al Saffar, H. Alkhayyat & K. Obaid (2024). A novel image encryption algorithm involving a logistic map and a self-invertible matrix. *Malaysian Journal of Mathematical Sciences*, *18*(1), 107–126. https://doi.org/10.47836/mjms.18.1.07.

[3] N. F. H. Al Saffar & M. R. M. Said (2015). Speeding up the elliptic curve scalar multiplication using the window-w non adjacent form. *Malaysian Journal of Mathematical Sciences*, *9*(1), 91–110.

[4] F. Al Shaarani & A. Gutub (2022). Securing matrix counting-based secret-sharing involving crypto steganography. *Journal of King Saud University-Computer and Information Sciences*, *34*(9), 6909–6924. https://doi.org/10.1016/j.jksuci.2021.09.009.

[5] N. H. M. Ali & S. A. Abead (2016). Modified Blowfish algorithm for image encryption using multi keys based on five sboxes. *Iraqi Journal of Science*, *57*(4C), 2968–2978.

[6] C. K. Chan & L. M. Cheng (2004). Hiding data in images by simple LSB substitution. *Pattern Recognition*, *37*(3), 469–474. https://doi.org/10.1016/j.patcog.2003.08.007.

[7] W. Diffie & M. E. Hellman (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, *22*(6), 644–654. https://doi.org/10.1109/TIT.1976.1055638.

[8] H. R. Hashim (2023). On the solutions of $2^x + 2^y = z^2$ in the Fibonacci and Lucas numbers. *Journal of Prime Research in Mathematics*, *19*(1), 27–33.

[9] H. R. Hashim & S. Tengely (2018). Representations of reciprocals of Lucas sequences. *Miskolc Mathematical Notes*, *19*(2), 865–872. https://doi.org/10.18514/MMN.2018.2520.

[10] E. S. B. Hureib & A. A. Gutub (2020). Enhancing medical data security via combining elliptic curve cryptography and image steganography. *International Journal of Computer Science and Network Security*, *20*(8), 1–8.

[11] E. S. B. Hureib & A. A. Gutub (2020). Enhancing medical data security via combining elliptic curve cryptography with 1-LSB and 2-LSB image steganography. *International Journal Computer Science and Network Security*, *20*(12), 232–241.

[12] S. A. Jebur, A. K. Nawar, L. E. Kadhim & M. M. Jahefer (2023). Hiding information in digital images using LSB steganography technique. *International Journal of Interactive Mobile Technologies*, *17*(7), 167–178. https://doi.org/10.3991/ijim.v17i07.38737.

[13] S. A. Jebur, A. K. Nawar, L. E. Kadhim & M. M. Jahefer (2023). Hiding information in digital images using LSB steganography technique. *International Journal of Interactive Mobile Technologies*, *17*(7), 167–178. https://doi.org/10.3991/ijim.v17i07.38737.

[14] P. Jindal & B. Singh (2015). RC4 encryption–A literature survey. *Procedia Computer Science*, *46*, 697–705. https://doi.org/10.1016/j.procs.2015.02.129.

[15] S. P. Kaur & S. Singh (2023). A digital steganography technique using hybrid encryption methods for secure communication. In *Proceedings of International Conference on Information Technology and Applications: ICITA 2022*, volume 614 pp. 481–489. Singapore. Springer. https://doi.org/10.1007/978-981-19-9331-2_41.

[16] N. Koblitz (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, *48*(177), 203–209. http://dx.doi.org/10.1090/S0025-5718-1987-0866109-5.

[17] E. J. Madarro Capó, C. M. Legón Pérez, O. Rojas & G. Sosa Gómez (2021). Information theory based evaluation of the RC4 stream cipher outputs. *Entropy*, *23*(7), Article ID: 896. https://doi.org/10.3390/e23070896.

[18] V. S. Miller (1985). Use of elliptic curves in cryptography. In *Conference On the Theory and Spplication of Cryptographic Techniques*, volume 218 pp. 417–426. Berlin, Heidelberg. Springer. https://doi.org/10.1007/3-540-39799-X_31.

[19] G. K. Murhty & T. Kanimozhi (2024). Methodologies in steganography and cryptography–review. In *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Volume 4*, pp. 205–214. Springer, Cham. https://doi.org/10.1007/978-3-031-43009-1_18.

[20] R. Panigrahi & N. Padhy (2025). An effective steganographic technique for hiding the image data using the LSB technique. *Cyber Security and Applications*, *3*, Article ID: 100069. https://doi.org/10.1016/j.csa.2024.100069.

[21] N. F. H. A. Saffar (2023). Steganography algorithm based on public key cryptosystem. In *The Second International Scientific Conference (SISC2021): College of Science, Al-Nahrain University*, volume 2457 pp. Article ID: 020011. Baghdad, Iraq. AIP Publishing. https://doi.org/10.1063/5.0118555.

[22] H. M. Salih & R. S. Al Mahdawi (2021). The security of RC4 algorithm using keys generation depending on user's retina. *Indonesian Journal of Electrical Engineering and Computer Science*, *24*(1), 452–463. http://doi.org/10.11591/ijeecs.v24.i1.pp452-463.

[23] B. Schneier (1993). Description of a new variable-length key, 64-bit block cipher (Blowfish). In *International workshop on fast software encryption*, volume 809 pp. 191–204. Berlin, Heidelberg. Springer. https://doi.org/10.1007/3-540-58108-1_24.

[24] N. P. Smart (2016). *Cryptography Made Simple*. Springer, Cham. https://doi.org/10.1007/978-3-319-21936-3.

[25] N. Subramanian, O. Elharrouss, S. Al-Maadeed & A. Bouridane (2021). Image steganography: A review of the recent advances. *IEEE Access*, *9*, 23409–23423. https://doi.org/10.1109/ACCESS.2021.3053998.

[26] O. L. Van Daalen (2023). The right to encryption: Privacy as preventing unlawful access. *Computer Law & Security Review*, *49*, Article ID: 105804. https://doi.org/10.1016/j.clsr.2023.105804.

[27] Y. G. Yang, B. P. Wang, Y. H. Zhou, W. M. Shi & X. Liao (2023). Efficient color image encryption by color-grayscale conversion based on steganography. *Multimedia Tools and Applications*, *82*(7), 10835–10866. https://doi.org/10.1007/s11042-022-13689-z.

[28] J. L. Yao, H. M. Yang, D. H. Jiang, B. Yan, J. S. Pan & M. X. Wang (2023). A novel quantum image steganography algorithm based on double-layer gray code. *International Journal of Theoretical Physics*, *62*(3), Article ID: 52. https://doi.org/10.1007/s10773-023-05303-1.